



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
AMAZONAS
CAMPUS MANAUS DISTRITO INDUSTRIAL
DEPARTAMENTO DE AUTOMAÇÃO E PRODUÇÃO
BACHARELADO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO**

GIOVANNI OLIVEIRA DE SOUSA

**TÉCNICA DE CONTROLE *FUZZY* TIPO-2: ESTUDO DE CASO DA
ESTABILIDADE E DESEMPENHO NA APLICAÇÃO DE ANESTESIA GERAL**

**MANAUS- AM
2022**

GIOVANNI OLIVEIRA DE SOUSA

**TÉCNICA DE CONTROLE *FUZZY* TIPO-2: ESTUDO DE CASO DA
ESTABILIDADE E DESEMPENHO NA APLICAÇÃO DE ANESTESIA GERAL**

Trabalho de Conclusão de Curso
apresentado ao IFAM - Campus Manaus
Distrito Industrial, Curso de Bacharelado
em Engenharia de Controle e Automação,
como requisito parcial para obtenção do
título de Bacharel em Engenharia de
Controle e Automação.

Orientador: Prof. Dr. Cleonor Crescêncio
das Neves

Coorientador: Prof. Dr. Ailton Gonçalves
Reis

**MANAUS-AM
2022**

Dados Internacionais de Catalogação na Publicação (CIP)

S725t Sousa, Giovanni Oliveira de.

Técnica de controle Fuzzy tipo-2: estudo de caso da estabilidade e desempenho na aplicação de anestesia geral. / Giovanni Oliveira de Sousa. – Manaus, 2022.

163f. : il. color

TCC (Graduação em engenharia de controle e automação) – Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, *Campus* Manaus Distrito Industrial, 2021.

Orientador: Prof. Dr. Cleonor Crescêncio das Neves

1. Simulação de sistema dinâmica. 2. Controle Fuzzy. 3. Controle Fuzzy tipo-2 intervalar. 4. Modelagem farmacocinética. 5. Controle de anestesia geral. I. Neves, Cleonor Crescêncio das (orient.) II. Instituto Federal de Educação, Ciência e Tecnologia do Amazonas. III. Título.

CDD 629.8

Elabora por Fc^a. Amélia Frota, registro n.858 (CRB11)

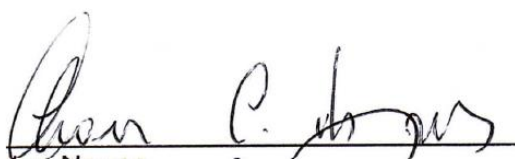
GIOVANNI OLIVEIRA DE SOUSA

**TÉCNICA DE CONTROLE FUZZY TIPO-2: ESTUDO DE CASO DA
ESTABILIDADE E DESEMPENHO NA APLICAÇÃO DE ANESTESIA GERAL**

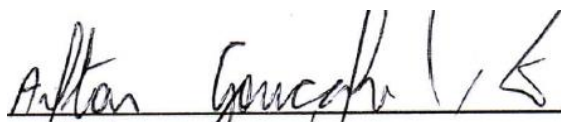
Trabalho de Conclusão de Curso
apresentado ao IFAM – Campus Manaus
Distrito Industrial, Curso de Bacharelado
em Engenharia de Controle e Automação,
como requisito parcial para obtenção do
título de Bacharel em Engenharia de
Controle e Automação.

Manaus, 29 de Abril de 2022

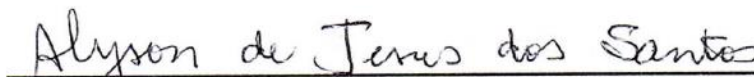
BANCA EXAMINADORA



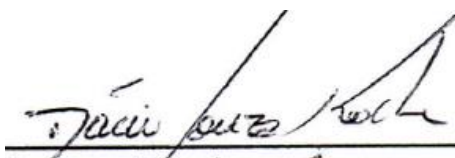
Orientador: Cleonor Crescêncio das Neves, Dr.



Coorientador: Ailton Gonçalves Reis, Dr.



Avaliador: Alyson de Jesus dos Santos, Dr.



Avaliador: Dário Souza Rocha, MSc.

**MANAUS-AM
2022**

Aos meus pais e minha irmã pelo
incentivo constante.

À minha namorada pela compreensão,
apoio e, sobretudo, pelo
companheirismo.

AGRADECIMENTOS

Gostaria de agradecer primeiramente aos meus pais, Marcílio e Shirleiane, pelo apoio e incentivo por toda a minha graduação e minha vida, até mesmo quando tinha pouca confiança que terminaria algum dia e quando não sabia se esse era verdadeiramente o meu caminho. Além de agradecer à minha irmã Giulia por nossa boa relação de irmãos por praticamente toda a minha vida.

Agradeço demais à minha amada Raissa, por tudo o que passamos e que nos tornamos juntos. Obrigado por uma relação comunicativa e amorosa para nós dois, mesmo com várias problemas em nossa frente. Você me ajudou a reencontrar o que realmente gosto na minha vida e te amo muito por tudo o que somos hoje.

E agradeço muito aos meus orientadores e amigos, professor Dr. Cleonor e professor Dr. Ailton, pela confiança de que eu poderia terminar e fazer um bom trabalho. Vou sempre ser grato por todas os conselhos, oportunidades e lições de vida e por apostarem em mim, me ajudando a me tornar um melhor aluno, profissional e pessoa.

Claro que não poderia ter conseguido passar pela faculdade sem os meus amigos mais queridos, Larissa, Beatriz e Guilherme, e por isso agradeço a vocês pela amizade acima de tudo e todas as nossas experiências. Nós nos apoiamos até o fim e acredito que conseguimos conquistar o que sempre acreditei que seríamos, não só formados e engenheiros, mas pessoas realizadas e felizes.

Por último, agradeço ao IFAM e ao corpo docente por todo o conhecimento passado, principalmente agradeço a vários professores que me ajudaram a entender que a engenharia é o meu caminho e se esforçaram para realmente ensinar a teoria e prática e preparar para, no final, me tornar um profissional competente.

RESUMO:

A anestesia geral depende da dosagem correta do medicamento no paciente e, conseqüentemente, o controle automático dessa dosagem se torna uma área de pesquisa importante e desafiadora. Com esse intuito, este trabalho investiga a aplicação da técnica do controle *fuzzy* tipo-2 na anestesia geral. Primeiramente, uma revisão bibliográfica foi realizada para o trabalho, estudando o controlador *fuzzy* tipo-1 e tipo-2 e sua aplicação na anestesia geral, além de procedimentos e técnicas utilizadas para compor um estudo de caso. Após esse estudo, foi implementado um modelo matemático farmacocinético farmacodinâmico do efeito dos medicamentos atracúrio e isoflurano no corpo humano. Para projetar os controladores *fuzzy* tipo-1 e tipo-2 intervalar foi utilizado o *Fuzzy Logic Toolbox* no ambiente MATLAB, em sequência, simulações do controle de anestesia foram criadas no MATLAB a partir de um modelo matemático. Em paralelo, um protótipo de baixa escala e baixo custo foi construído para representar o sistema de controle anestésico real. Os resultados da simulação mostraram a superioridade dos controladores *fuzzy* tipo-2 intervalar em relação aos do tipo-1 com o mesmo número de entradas, entretanto foi possível verificar que o maior número de entradas beneficiou tanto os controles com tipo-2 intervalar e tipo-1. E por fim, foi realizada uma análise comparativa e foi obtido que a lógica *fuzzy* tipo-2 é uma boa alternativa para controlar sistemas multivariáveis e não lineares.

Palavras-chave: Simulação de Sistemas Dinâmicos, Controle *Fuzzy*, Controle *Fuzzy* tipo-2 intervalar, Modelagem farmacocinética farmacodinâmica e Controle da anestesia geral.

ABSTRACT:

General anesthesia depends on the correct dosage on the patient and, consequentially, the automatic control of this dosage becomes an important and challenging research field. For that propose, this work investigates the application of the type-2 fuzzy control's technique in general anesthesia. Firstly, a literature review was made for the work, studying type-1 and interval type-2 fuzzy controllers and their application on general anesthesia to compose case study. After this study, a pharmacokinetic/pharmacodynamic mathematical model of the effect of the drugs Atracurium and Isoflurane in the human body was implemented. To design the Type-1 and interval type-2 fuzzy controllers, the Fuzzy Logic Toolbox was used in the MATLAB environment, and then, anesthesia control simulations were made on MATLAB starting from a mathematical model. At the same time, a small-scale, low-cost prototype was built to represent the real anesthesia control system. Simulation results showed type-2 fuzzy controllers' superiority against type-1 with the same number of inputs, however it was possible to see that the bigger number of inputs were beneficial for type-2 control as well as type-1. And finally, a comparative analysis was made, thus the conclusion was that type-2 fuzzy logic is a good alternative to control multivariable and nonlinear systems.

Keywords: Dynamic Systems Modelling. Fuzzy Control. Interval Type-2 Fuzzy Control. Pharmacokinetic/Pharmacodynamic Modelling. Control of General Anesthesia.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de lógica <i>fuzzy</i> utilizando valores linguísticos tradicionais	26
Figura 2 – Comparação entre conjuntos <i>fuzzy</i> tipo-1 e tipo-2	27
Figura 3 – Função de pertinência discreta <i>fuzzy</i> genérica tipo-2	28
Figura 4 – Função de pertinência discreta <i>fuzzy</i> tipo-2 intervalar	29
Figura 5 – Estrutura básica do controlador <i>fuzzy</i>	30
Figura 6 – Estrutura básica do controlador <i>fuzzy</i> tipo-2	35
Figura 7 – Modelo Farmacocinético com dois compartimentos	40
Figura 8 – Modelo PK/PD com dois compartimentos.....	41
Figura 9 – Modelo PK/PD em diagrama de blocos	42
Figura 10 – Modelo PK/PD do Atracúrio em diagrama de blocos.....	43
Figura 11 – Modelo PK/PD do Isoflurano em diagrama de blocos.....	44
Figura 12 – Modelo PK/PD do corpo humano na anestesia em diagrama de blocos.....	45
Quadro 1 – Equipamentos utilizados no trabalho.....	49
Figura 13 – Proposta do trabalho em fluxograma	51
Figura 14 – Valores linguísticos das entradas dos controladores T1.FLC	53
Figura 15 – Valores linguísticos das saídas dos controladores T1.FLC.....	54
Quadro 2 – Estrutura das bases de regras dos controladores com duas entradas	55
Quadro 3 – Base de regras para a saída Atr.inf no controlador com duas entradas	55
Quadro 4 – Base de regras para a saída Iso.conc no controlador com duas entradas.....	56
Quadro 5 – Estrutura das bases de regras dos controladores com quatro entradas	57
Quadro 6 – Base de regras para a saída Atr.inf no controlador com quatro entradas	57
Quadro 7 – Base de regras para a saída Iso.conc no controlador com quatro entradas..	58
Figura 16 – Diagrama funcional do controlador T1.22.FLC.....	60
Figura 17 – Diagrama de decomposição do controlador T1.22.FLC	61
Figura 18 – Diagrama funcional do controlador T1.42.FLC.....	62
Figura 19 – Diagrama de decomposição do controlador T1.42.FLC	63
Figura 20 – Valores linguísticos da entrada RM.e dos controladores IT2.FLC.....	65
Figura 21 – Valores linguísticos da saída Atr.inf dos controladores IT2.FLC.....	66
Figura 22 – Diagrama funcional do controlador IT2.22.FLC	68
Figura 23 – Diagrama de decomposição do controlador IT2.22.FLC.....	69
Figura 24 – Diagrama funcional do controlador IT2.42.FLC	70
Figura 25 – Diagrama de decomposição do controlador IT2.42.FLC.....	71

Figura 26 – Resultado da simulação do controlador T1.22.FLC.....	74
Figura 27 – Resultado da simulação do controlador T1.42.FLC.....	75
Figura 28 – Resultado da simulação do controlador IT2.22.FLC	76
Figura 29 – Resultado da simulação do controlador IT2.42.FLC	77
Gráfico 1 – Comparação gráfica entre os resultados em Relaxamento Muscular.....	78
Gráfico 2 – Comparação gráfica entre os resultados em Pressão Sanguínea	79
Quadro 8 – Comparação da utilização dos equipamentos no protótipo	83
Figura 30 – Protótipo do sistema montado	84

LISTA DE TABELAS

Tabela 1 – Comparação dos modelos de inferência de controladores <i>fuzzy</i>	33
Tabela 2 – Parâmetros das funções de pertinências para a fuzzificação	54
Tabela 3 – Parâmetros das funções de pertinências para a defuzzificação	55
Tabela 4 – Escalares utilizados nos controladores T1.FLC	59
Tabela 5 – Parâmetros das funções de pertinência das entradas dos IT2.FLC.....	66
Tabela 6 – Parâmetros das funções de pertinência das saídas dos IT2.FLC	67
Tabela 7 – Escalares utilizados nos controladores IT2.FLC	67
Tabela 8 – Parâmetros das simulações	72
Tabela 9 – Comparação numérica entre os resultados em Relaxamento Muscular	79
Tabela 10 – Comparação numérica entre os resultados em Pressão Sanguínea.....	80

LISTA DE ABREVIATURAS E SIGLAS

ADC – *Analog-to-Digital Converter*

Atr.inf – Infusão do Atracúrio

BIS – *Bispectral Index*

BP – *Blood Pressure*

COA – *Center Of Area*

DC – *Direct Current*

DoA – *Depth of Anesthesia*

EEG – Eletroencefalografia

EG – Erro Grande

EIASC – *Enhanced Iterative Algorithm with Stopping Condition*

EKM – *Enhanced Karnik-Mendel*

EM – Erro Médio

EMG – Eletromiografia

EP – Erro Pequeno

FOU – *Footprint Of Uncertainty*

GA – *Genetic Algorithm*

GP – *Genetic Programming*

i.PS.e – Integral do Erro na Pressão Sanguínea

i.RM.e – Integral do Erro no Relaxamento Muscular

Iso.conc – Concentração do Isoflurano

IT2.22.FLC – Controlador IT2.FLC com duas entradas e duas saídas

IT2.42.FLC – Controlador IT2.FLC com quatro entradas e duas saídas

IT2.FLC – *Interval Type-2 Fuzzy Logic Controller*

IWA – *Interval Weighted Average*

KM – *Karnik-Mendel*

MAP – *Mean Arterial Pressure*

MF – *Membership Function*

MIMO – *Multiple Inputs, Multiple Outputs*

MPC – *Model Predictive Control*

NB – *Negative Big*

NM – *Negative Medium*

NS – *Negative Small*

NVPO – Náusea e Vômito no Pós-Operatório

PB – *Positive Big*

PCB – *Printed Circuit Board*

PK/PD – *Pharmacokinetics/Pharmacodynamics*

PM – *Positive Medium*

PS – *Positive Small*

PS – Pressão Sanguínea

PS.e – Erro na Pressão Sanguínea

PWM – *Pulse Width Modulation*

RM – Relaxamento Muscular

RM.e – Erro no Relaxamento Muscular

RNA – Rede Neural Artificial

sFLC – Sub-Controlador *Fuzzy*

SMC – *Sliding Mode Control*

SOFLC – *Self-Organizing Fuzzy Logic Controller*

T1.22.FLC – Controlador T1.FLC com duas entradas e duas saídas

T1.42.FLC – Controlador T1.FLC com quatro entradas e duas saídas

T1.FLC – *Type-1 Fuzzy Logic Controller*

T2SOFLC – *Type-2 Self-Organizing Fuzzy Logic Controller*

TS – Tagaki-Sugeno

ZE – *Zero*

LISTA DE SÍMBOLOS

X – Universo de discurso

A – Conjunto *crisp*

f_A – Função característica

$\underline{A}$ – Conjunto *fuzzy* tipo-1

$\mu_{\underline{A}}$ – Função de pertinência para o conjunto $\underline{A}$

x – Elemento do universo de discurso

$\mu_{\underline{A}}(x)$ – Função de pertinência de x para o conjunto $\underline{A}$

$\underline{B}$ – Conjunto *fuzzy* tipo-1 distinto de $\underline{A}$ no mesmo universo de discurso

$\mu_{\underline{A} \cup \underline{B}}(x)$ – União dos conjuntos $\underline{A}$ e $\underline{B}$

$\mu_{\underline{A} \cap \underline{B}}(x)$ – Interseção dos conjuntos $\underline{A}$ e $\underline{B}$

$\mu_{\underline{A}^c}(x)$ – Complemento do conjunto $\underline{A}$

a – Parâmetro de início da função de pertinência triangular

b – Parâmetro de pico da função de pertinência triangular

c – Parâmetro de fim da função de pertinência triangular

$\tilde{\underline{A}}$ – Conjunto *fuzzy* tipo-2

J_x – Faixa de valores de pertinência de x

u – Valor final de J_x

$\mu_{\tilde{\underline{A}}}(x, u)$ – Função de pertinência de x e u para o conjunto $\tilde{\underline{A}}$

t – Tempo

$e(t)$ – Erro em função de t medido entre a referência e o sensor

m – Número total de regras

n – Número total de variáveis

i – Número atual da variável

j – Número atual da regra

$A_{i,j}$ – Valor linguístico de uma regra

y_j – Ação de uma regra

C_j – Conjunto de uma regra

R_j – Estrutura genérica de uma regra

x_n – Variável n

x_i – Variável definida no universo X_i

x_i^* – Valor instantâneo da variável x_i

$\alpha_{i,j}$ – compatibilidade de x_i^* com $A_{i,j}$ com a regra R_j

$\top$ – T-norma

α_j – Compatibilidade global da regra com os valores instantâneos x_i^*

C_j' – Novo conjunto de C_j relacionado com α_j

I – Implicação

$\mu_{C_j'}(y)$ – Função de pertinência de C_j' relacionado a y

∇ – Agregação

$\mu_{C'}(y)$ – Função de pertinência do conjunto final C' a partir de todos os C_j'

u^* – Valor para a saída final

N – Número total de valores linguísticos da saída

u_i – Valor de posição de centroide de cada área individual

$\mu_{OUT}(u_i)$ – Pertinência correspondente de u_i

k_{10} – Parâmetro de eliminação do remédio do compartimento 1

k_{20} – Parâmetro de eliminação do remédio do compartimento 2

k_{12} – Constante de movimentação do remédio do compartimento 1 para o 2

k_{21} – Constante de movimentação do remédio do compartimento 2 para o 1

$\dot{x}_1$ – Variação da concentração do remédio no compartimento 1

v_1 – Volume do compartimento 1

x_E – Concentração do remédio no compartimento efeito

v_E – Volume do compartimento efeito

k_{1E} – Constante de movimentação do remédio do compartimento 1 para o efeito

$\dot{x}_E$ – Variação da concentração do remédio no compartimento efeito

E – Efeito

E_{max} – Efeito máximo

$x_E(50)$ – Efeito com concentração em 50%

α – Constante para a curva sigmoide

E_0 – Efeito nulo

$U_{(inf)}(s)$ – Infusão genérica do Atracúrio

$G_{PK-PD_{(atr)}}(s)$ – Função de transferência do modelo PK/PD do Atracúrio

$X_E(s)$ – Concentração do efeito

$E_{(atr)}$ – Efeito do atracúrio

$U_{(conc)}(s)$ – Concentração genérica do Isoflurano

$G_{PK-PD_{(iso)}}(s)$ – Função de transferência do modelo PK/PD do Isoflurano

$\Delta MAP(s)$ – Variação da pressão média arterial

$U_1(s)$ – Infusão do Atracúrio no modelo do corpo humano na anestesia

$U_2(s)$ – Concentração do Isoflurano no modelo do corpo humano na anestesia

$RM(s)$ – Efeito de Relaxamento Muscular

$PS(s)$ – Efeito de Pressão Sanguínea

$G_{11}(s)$ – Função de transferência do Atracúrio no Relaxamento Muscular

$G_{22}(s)$ – Função de transferência do Isoflurano na Pressão Sanguínea

$G_{12}(s)$ – Função de transferência do Isoflurano no Relaxamento Muscular

$n!$ – Número total de elementos em um combinação

$k!$ – Número de elementos a serem combinados

C_k^n – Combinação simples de n em k

C_2^4 – Combinação simples de 4 em 2

$\vee$ – Operador que retorna o valor máximo

SUMÁRIO

1 INTRODUÇÃO	19
2 REVISÃO BIBLIOGRÁFICA	22
2.1 LÓGICA <i>FUZZY</i>	22
2.1.1 Conjuntos <i>Crisp</i>	23
2.1.2 Conjuntos <i>Fuzzy</i>	23
2.1.3 Operações Em Conjuntos <i>Fuzzy</i>	24
2.1.3.1 União	24
2.1.3.2 Interseção.....	25
2.1.3.3 Complemento	25
2.1.4 Variável Linguística e Valor Linguístico.....	25
2.1.5 Função de Pertinência.....	26
2.1.6 Conjuntos <i>Fuzzy</i> Tipo-2	27
2.1.6.1 Genérico	28
2.1.6.2 Intervalar.....	28
2.2 CONTROLE <i>FUZZY</i> BASEADO EM REGRAS	29
2.2.1 Fuzzificação.....	30
2.2.2 Base de Conhecimento.....	31
2.2.3 Inferência.....	32
2.2.4 Defuzzificação	33
2.2.5 Controle <i>Fuzzy</i> Tipo-2.....	33
2.2.5.1 Tipo-redutor.....	35
2.2.6 Controladores <i>Fuzzy</i>	36
2.3 ANESTESIA GERAL	36
2.3.1 Definição da Anestesia.....	37
2.3.2 Procedimentos e Sinais Biomédicos Utilizados	37
2.3.3 Erro de Dosagem na Anestesia	38
2.3.4 Medicamentos Utilizados na Anestesia.....	39
2.3.4.1 Farmacocinética.....	40
2.3.4.2 Farmacodinâmica.....	40
2.3.4.3 Modelo PK/PD	42
2.3.4.4 Atracúrio.....	42
2.3.4.5 Isoflurano.....	43

2.3.4.6 Modelo Multivariável do Corpo Humano para Anestesia.....	44
2.3.5 Controle da Anestesia Geral.....	46
2.3.5.1 Controle <i>Fuzzy</i> da Anestesia.....	46
2.3.5.2 Outros Tipos de Controle da Anestesia	47
2.3.5.3 Proposta do Trabalho.....	48
3 MATERIAIS E MÉTODOS	49
3.1 MATERIAIS UTILIZADOS.....	49
3.2 METODOLOGIA DO TRABALHO	50
3.3 PROJETO DOS CONTROLADORES <i>FUZZY</i> TIPO-1 (T1.FLC)	52
3.3.1 Especificações de projeto dos controladores T1.FLC	52
3.3.2 Controlador com duas entradas e duas saídas (T1.22.FLC)	59
3.3.3 Controlador com quatro entradas e duas saídas (T1.42.FLC).....	61
3.4 PROJETO DE CONTROLADORES <i>FUZZY</i> TIPO-2 INTERVALAR (IT2.FLC) 64	
3.4.1 Especificações de projeto dos controladores IT2.FLC.....	64
3.4.2 Controlador com duas entradas e duas saídas (IT2.22.FLC).....	67
3.4.3 Controlador com quatro entradas e duas saídas (IT2.42.FLC)	69
4 SIMULAÇÕES E ANÁLISES DOS RESULTADOS	72
4.1 SIMULAÇÕES.....	72
4.1.1 Simulação do controlador T1.22.FLC	73
4.1.2 Simulação do controlador T1.42.FLC	75
4.1.3 Simulação do controlador IT2.22.FLC.....	76
4.1.4 Simulação do controlador IT2.42.FLC.....	76
4.1.5 Comparação das simulações	77
4.2 ANÁLISE DOS RESULTADOS	80
4.3 PROTÓTIPO REPRESENTATIVO DO SISTEMA	83
5 CONSIDERAÇÕES FINAIS.....	86
5.1 TRABALHOS FUTUROS	87
REFERÊNCIAS	88

1 INTRODUÇÃO

De acordo com Doctor *et al.* (2016), normalmente os procedimentos cirúrgicos modernos utilizam a anestesia geral, a qual pode ser feita com fármacos anestésicos intravenosos ou inalatórios. Doses aumentadas dos medicamentos, de acordo com Amiri *et al.* (2020) podem causar náusea e vômitos no pós-operatório (NVPO).

Ilyas *et al.* (2017) discute sobre os benefícios da medicina moderna com relação aos procedimentos cirúrgicos, especificamente o conforto e a facilidade, citando o desenvolvimento da pesquisa da anestesia como um dos principais fatores que possibilitaram tal fato.

Uma das principais funções do médico anestesista, de acordo com Doctor *et al.* (2016), consiste na manutenção da anestesia por meio dosagem precisa de medicamentos anestésicos. Sharma *et al.* (2020) comenta que é necessário tanto no procedimento cirúrgico quanto pós-cirurgia o monitoramento contínuo da introdução de fármacos para não causar nenhum mal ao paciente. Geralmente uma rotina específica de administração de fármacos está definida durante diferentes estágios do procedimento, dependendo de fatores como idade ou peso do paciente (GARG *et al.*, 2019).

Ainda existem alguns problemas que podem ocorrer quando um medicamento não precisamente dosado chega ao paciente, como a retomada de lucidez do paciente ainda no procedimento cirúrgico ou recuperação da cirurgia prolongada (DOCTOR *et al.*, 2016).

O controle ou regulação dos fármacos no procedimento anestésico geralmente continua manual, ou seja, a aplicação do medicamento se dá pelo médico ou enfermeiro responsável, assim como os possíveis erros de dosagem que podem causar alguns problemas. Esse controle poderia ser feito de maneira automática, assim como outros casos similares, porém, existem características especiais nesse problema que dificultam a utilização de controladores convencionais que são utilizados na indústria, como a dinâmica de variação de parâmetros intrapaciente, entre dois pacientes diferentes, e interpaciente, no mesmo paciente em tempos diferentes (ILYAS *et al.*, 2017). Assim, a pergunta problema que norteia a pesquisa se torna: Como aplicar as técnicas de controle *fuzzy* na dosagem dos medicamentos da anestesia geral, com o intuito de dar maior eficiência e segurança nos cuidados do paciente?

Atualmente há pesquisas na utilização de controladores mais robustos no problema de controle da anestesia como em Ilyas *et al.* (2017) e Wei *et al.* (2020). Como

exemplo, pode-se citar o controlador *fuzzy* que tem uma facilidade maior em resolver problemas complexos de controle multivariável e não linear. Esse sistema baseado em lógica *fuzzy*, também chamada de tipo-1, consegue levar em consideração a incerteza que existe na medicina, e, conseqüentemente, nas técnicas de dosagem anestésica. Tipos específicos desses, como o controlador *fuzzy* autoajustável (SOFLC, do inglês *Self-Organizing Fuzzy Logic Controller*) ou controlador *fuzzy* tipo-2 autoajustável (T2SOFLC, do inglês *Type-2 Self-Organizing Fuzzy Logic Controller*) tiveram resultados promissores em simulações que mostraram até maior robustez em relação aos controladores *fuzzy* tradicionais (SHIEH *et al.*, 2009; Doctor *et al.*, 2016).

Dessa forma, um sistema automático baseado em controle *fuzzy* será útil a partir de dados biomédicos anestésicos e experiência e conhecimento de especialistas na área para um melhor desempenho do controle de anestesia.

Particularmente, a lógica *fuzzy* tipo-2 se torna uma ferramenta importante para essa área de estudo porque ela avalia em torno de uma função de pertinência com uma margem de erro, entre a função de pertinência máxima e mínima, dessa forma, esse trabalho acadêmico torna-se relevante por procurar mostrar a capacidade da lógica *fuzzy* em resolver problemas complexos em prol da sociedade e, portanto, sua realização é justificada.

Assim, o objetivo geral deste trabalho foi: Aplicar a técnica de controle *fuzzy* tipo-2 na anestesia geral. Desse objetivo geral decorrem cinco específicos, quais sejam:

- a) estudar os conceitos pertinentes para a realização da pesquisa;
- b) abordar problemas existentes em caso de erro de dosagem de medicamento em anestesia geral;
- c) identificar as técnicas de controle automático *fuzzy* tipo-1 e tipo-2 aplicadas na anestesia geral;
- d) projetar sistemas automáticos de controle *fuzzy* tipo-1 e tipo-2 para a aplicação na anestesia geral;
- e) verificar e comparar o desempenho dos sistemas por meio de simulação.

Os referenciais teóricos neste trabalho evidenciam, dentre outros, trabalhos com a mesma temática como em Doctor *et al.* (2016) e Shieh *et al.* (2009) e levaram a estudos aprofundados de temas mais gerais como controle *fuzzy* e anestesia geral.

A metodologia do trabalho respeita as características de uma pesquisa qualitativa, aplicada, explicativa e bibliográfica.

Os resultados mostraram que a técnica de controle *fuzzy* tipo-2 é superior à de tipo-1 e é uma opção viável para o controle da anestesia geral.

Espera-se que este trabalho contribua para o desenvolvimento de tecnologias de controle utilizando a lógica *fuzzy* tipo-2 e para a mitigação de erros de dosagem na anestesia geral proporcionando menores riscos ao paciente no procedimento.

Este trabalho tem como principal foco o tratamento de técnica de controle *fuzzy* tipo-2 para anestesia. Dessa forma, será dividido em cinco capítulos.

No capítulo 1, foi introduzido o tema, problemática, justificativa, objetivos e contribuições esperados do trabalho.

No capítulo 2, será abordada uma revisão bibliográfica sobre a lógica *fuzzy* tipo-1 e tipo-2; a estrutura do controle *fuzzy* tipo-1 e tipo-2, além da menção de vários controladores diferentes encontrados na bibliografia; o processo da anestesia geral, os procedimentos e sinais utilizados, erros de dosagem, alguns medicamentos utilizados e suas características e algumas formas de controle automático encontrados na bibliografia.

No capítulo 3, descreve-se os materiais e métodos do trabalho.

No capítulo 4, serão apresentadas e discutidas as simulações e análise dos resultados.

No capítulo 5, mostra-se as considerações finais e possíveis trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo serão apresentados conceitos de temas pertinentes ao trabalho, sendo divididos em estudos sobre Lógica *Fuzzy*, Controle *Fuzzy* e Anestesia Geral.

2.1 LÓGICA *FUZZY*

A teoria clássica dos conjuntos avalia elementos que se encaixam dentro de certos parâmetros, sejam numéricos e/ou lógicos, formando conjuntos, nos quais tais elementos podem pertencer ou não a um determinado conjunto (BERNADES, 2019). Porém, um problema aparece quando esses elementos não tiverem uma classificação tão precisa em um conjunto.

Desenvolvido por Zadeh em 1965, a teoria de conjuntos *fuzzy* e lógica *fuzzy*, difusa ou nebulosa propõe modelar matematicamente a linguagem comum, levando em consideração sua incerteza, ambiguidade, imprecisão e subjetividade (GUILLET, 2019). Alguns elementos poderiam pertencer mais ou serem mais pertinentes do que outros em diferentes níveis a conjuntos, a partir dessa ideia a lógica *fuzzy* foi criada (BERNADES, 2019).

De acordo com Sandri e Correa (1999), a imprecisão é intrinsicamente ligada e inversamente proporcional à incerteza, ou seja, quanto maior for a primeira, menor a segunda. As principais teorias que tratam com a imprecisão e incerteza são, respectivamente, a teoria de conjuntos e da probabilidade, porém, elas nem sempre conseguem traduzir a rica informação oferecida por seres humanos. Assim, a teoria dos conjuntos *fuzzy* se propõe a abordar o aspecto vago da informação humana e, se for utilizada em situações lógicas, como sistemas baseados em conhecimento, é conhecida como lógica *fuzzy*.

A comunicação do ser humano possui expressões que são, diversas vezes, vagas ou imprecisas (SIMÕES; SHAW, 2007). Por exemplo, uma pessoa pode ser questionada se sua refeição está apimentada, sua resposta pode ser além de um simples sim ou não. A pergunta em si é vaga, já que poderia ter sido especificado se a comida está apimentada para o paladar da pessoa ou mais genericamente para qualquer outra. Além disso, a resposta pode apresentar uma variedade de modificadores de predicado que geram termos linguísticos (GOMIDE; GUDWIN; TANSCHIEDT, 1995), nesse caso, poderia dizer que a comida está muito ou pouco apimentada, ou apimentada demais, com ênfase nos

modificadores muito e pouco, já que eles indicam a incerteza ou imprecisão da informação.

A lógica *fuzzy* pode ser utilizada para, até um certo ponto, permitir que um computador processe informações imprecisas, relacionadas à comunicação humana, utilizando seus graus de verdade, numericamente é dado no intervalo $[0,1]$, ou seja, raciocinar que nem um ser humano (SIMÕES; SHAW, 2007).

Nos seguintes tópicos, serão abordadas as mais importantes características teóricas da lógica *fuzzy*, incluindo a teoria dos conjuntos *fuzzy* que é considerada uma extensão da teoria clássica dos conjuntos.

2.1.1 Conjuntos *Crisp*

Se limitando a um universo de discurso X , podendo ser discreto ou contínuo, um conjunto de definição clássica ou *crisp* é uma coleção de elementos x dentro desse universo, enquanto um subconjunto é uma coleção dentro do conjunto (SIVANANDAM; SUMATHI; DEEPA, 2007).

O mecanismo utilizado para identificar se um elemento pertence ou não pertence a um conjunto em um universo é chamado de pertinência. A função matemática formal para a pertinência é chamada de função característica, sendo escrita como, para o conjunto A no universo de discurso X (AGUIRRE *et al.*, 2007):

$$f_A: X \rightarrow 0,1 \quad (1)$$

Tal que:

$$f_A(x) = \begin{cases} 1, & \text{se e somente se } x \in A \\ 0, & \text{caso contrário} \end{cases} \quad (2)$$

Cada elemento x do universo de discurso X terá um dos dois valores possível quanto a sua pertinência ao conjunto, ou seja, poderá pertencer ou não pertencer.

2.1.2 Conjuntos *Fuzzy*

De acordo com Aguirre *et al.* (2007), um conjunto *fuzzy* é baseado na função de pertinência μ_A , que pode assumir infinitos valores entre 0 e 1, ou seja, para o conjunto *fuzzy* $\underline{A}$ e um universo X :

$$\mu_{\underline{A}}: X \rightarrow [0,1] \quad (3)$$

Os valores possíveis da função de pertinência permitem a capacidade de incluir a incerteza quanto a um elemento pertencer a um conjunto, já que quanto maior sua pertinência, terá maior certeza de que realmente pertence. A ideia de poder incluir elementos que não possuem pertinência completa com o conjunto, quando for igual a 1, entra em contraste com os conjuntos clássicos que não permitiriam a entrada (SIVANANDAM; SUMATHI; DEEPA, 2007). Assim, são descritos conjuntos mais complexos de uma maneira mais humana, com dúvidas e incertezas.

Conjuntos *fuzzy* têm uma função de pertinência μ_A como característica principal, assim são representados por um conjunto de pares ordenados, cada um representando a pertinência e o elemento do conjunto (GOMIDE; GUDWIN; TANSCHIEDT, 1995), para um conjunto *fuzzy* $\underline{A}$:

$$\underline{A} = \mu_A(x)/x \quad (4)$$

O universo de um conjunto *fuzzy* pode ser contínuo ou discreto. De acordo com Aguirre *et al.* (2007), caso for contínuo, o conjunto *fuzzy* normalmente pode ser representado pela união dos pares ordenados, de maneira diferente de uma integral usual:

$$\underline{A} = \int_x \mu_A(x)/x \quad (5)$$

2.1.3 Operações Em Conjuntos Fuzzy

Assim como em conjuntos clássicos, existe a possibilidade de operações matemáticas entre conjuntos *fuzzy*, ou seja, as operações clássicas de conjuntos (como união ou interseção de dois ou mais conjuntos) podem ser estendidas. As principais serão especificadas nas próximas seções definindo $\underline{A}$ e $\underline{B}$ como dois conjuntos *fuzzy* distintos no mesmo universo de discurso X (SIVANANDAM; SUMATHI; DEEPA, 2007).

2.1.3.1 União

De maneira similar aos conjuntos clássicos, a união de dois ou mais conjuntos *fuzzy* forma outro conjunto, esse contém elementos pertencentes a pelo menos um dos conjuntos originais (AGUIRRE *et al.*, 2007).

Porém, enquanto a união de conjuntos clássicos só pode incluir mais elementos, a operação com conjuntos *fuzzy* também poderá aumentar o grau de pertinência utilizando o valor máximo, caso o mesmo elemento estiver nos dois conjuntos. Assim, a união

poderá ser representada de duas formas idênticas (SIVANANDAM; SUMATHI; DEEPA, 2007):

$$\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x) \quad (6)$$

$$\mu_{A \cup B}(x) = \max(\mu_A(x); \mu_B(x)) \quad (7)$$

2.1.3.2 Interseção

Para conjuntos clássicos, a interseção de dois ou mais conjuntos *fuzzy* também forma outro conjunto, esse contém elementos que existem em todos os conjuntos (AGUIRRE *et al.*, 2007). Porém, enquanto a interseção de conjuntos clássicos só pode incluir ou excluir elementos, a operação com conjuntos *fuzzy* também poderá diminuir o grau de pertinência utilizando o valor mínimo de pertinência para o mesmo elemento. Assim, a interseção poderá ser representada de duas formas idênticas (SIVANANDAM; SUMATHI; DEEPA, 2007):

$$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) \quad (8)$$

$$\mu_{A \cap B}(x) = \min(\mu_A(x); \mu_B(x)) \quad (9)$$

2.1.3.3 Complemento

O complemento de um conjunto *fuzzy* será outro conjunto contendo o inverso da função de pertinência de cada elemento (SIVANANDAM; SUMATHI; DEEPA, 2007):

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (10)$$

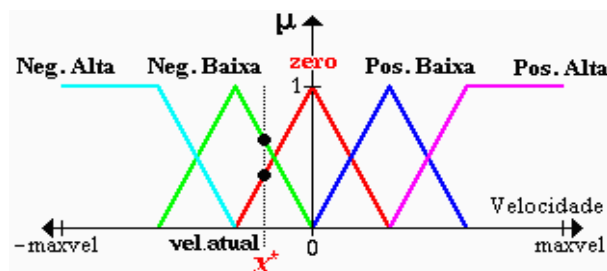
2.1.4 Variável Linguística e Valor Linguístico

De acordo com Aguirre *et al.* (2007), uma variável linguística tem seus valores representados por termos linguísticos relacionados à comunicação humana. Esses valores são representados por conjuntos *fuzzy*, portanto há uma distinção entre a variável e o valor.

Os valores linguísticos podem ser construídos a partir de termos primários (como *rápido*), conectivos lógicos (como *não* e *ou*) e modificadores (como *muito* ou *pouco*) (AGUIRRE *et al.*, 2007). Assim, existem várias combinações diferentes de termos linguísticos usados na lógica *fuzzy*.

De acordo com Simões e Shaw (2007), os valores linguísticos tradicionais em algum sistema industrial são somente sete: NB (grande negativo, do inglês *Negative Big*); NM (médio negativo, do inglês *Negative Medium*); NS (pequeno negativo, do inglês *Negative Small*); ZE (zero ou próximo de zero, , do inglês *Zero*); PS (pequeno positivo, do inglês *Positive Small*); PM (médio positivo, do inglês *Positive Medium*); PB (grande positivo, do inglês *Positive Big*). Todos os valores estão escritos em inglês. Esses valores fazem sentido especialmente para o controle de processos, já que tradicionalmente é feito a partir do erro positivo ou negativo entre a referência e valor real da variável controlada. A Figura 1 abaixo demonstra um exemplo com cinco valores dos sete:

Figura 1 – Exemplo de lógica *fuzzy* utilizando valores linguísticos tradicionais



Fonte: Sandri e Correa (1999)

2.1.5 Função de Pertinência

A função de pertinência (MF – *Membership Function*) indica o valor de pertinência, representado pelo símbolo $\mu_A(x)$, no sentido do elemento x pertencer ao conjunto A caracterizado por uma função matemática. O valor 0 significa que existe uma certeza de que o elemento não pertence ao conjunto, enquanto o valor 1 indica que o elemento pertence totalmente ao conjunto.

A escolha da função de pertinência é algo diferente para cada pessoa, os tipos mais comuns são de forma triangular, trapezoidal e Gaussiana, mas podem mudar de forma e parâmetros em uma aplicação prática (AGUIRRE *et al.*, 2007). Um exemplo é a função triangular, como mostrado na Figura 1 nos valores NB, ZE e PB, tendo os parâmetros genéricos a , b e c que são, respectivamente, o início, pico e fim da função para o elemento x pela Equação 11 a seguir (AZAM *et al.*, 2020):

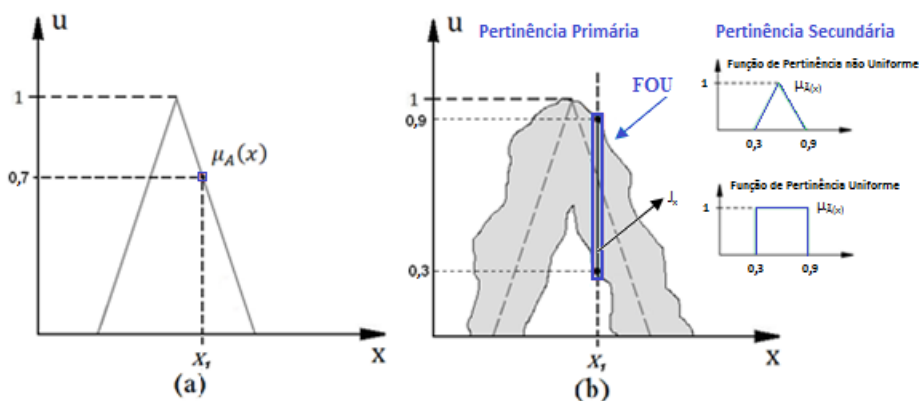
$$\mu_A(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \quad (11)$$

2.1.6 Conjuntos Fuzzy Tipo-2

Os conjuntos *fuzzy* representam variáveis linguísticas, em que seu significado representa incerteza, de maneira precisa (RIZOL; MESQUITA; SAOTOME, 2011). De acordo com os mesmos autores, os conjuntos *fuzzy* tipo-2 modelam a incerteza do significado das palavras utilizando uma “mancha” de incerteza, chamada de FOU (*Footprint Of Uncertainty*), ou seja, estende a função do *fuzzy* convencional, que será diferenciado no restante do trabalho como *fuzzy* tipo-1.

A diferença entre os dois conjuntos pode ser analisada na Figura 2 (RIZOL; MESQUITA; SAOTOME, 2011). Basicamente, o *fuzzy* tipo-1 utiliza sua função de pertinência $\mu_A(x)$ para transformar um valor real x em uma única pertinência 0,7, enquanto o tipo-2 transforma para uma faixa de valores *fuzzy* de pertinência $x = [0,3; 0,9]$, ou seja, dois níveis de pertinência são utilizados, a primária e a secundária, com a possibilidade de ser não uniforme (genérico) ou uniforme (intervalar).

Figura 2 – Comparação entre conjuntos *fuzzy* tipo-1 e tipo-2



Fonte: Rizol; Mesquita e Sautome (2011)

A principal diferença do conjunto *fuzzy* tipo-2 em relação ao tipo-1 está na ideia de que o próprio conjunto é *fuzzy*, não *crisp*, ou seja, não existe um único valor definido no conjunto *fuzzy* para um certo valor real. Dessa forma, é possível modelar incertezas em que o *fuzzy* tipo-1 não consegue, como medições com ruídos ou variáveis linguísticas que têm significado diferente para diferentes pessoas (MENDEL, 2017).

Nas próximas seções são apresentados os principais tipos de conjuntos *fuzzy* tipo-2, além de suas vantagens e desvantagens.

2.1.6.1 Genérico

De acordo com Rizol, Mesquita e Saotome (2011), um conjunto *fuzzy* tipo-2 é caracterizado pela função de pertinência $\mu_{\tilde{A}}(x, u)$ em que:

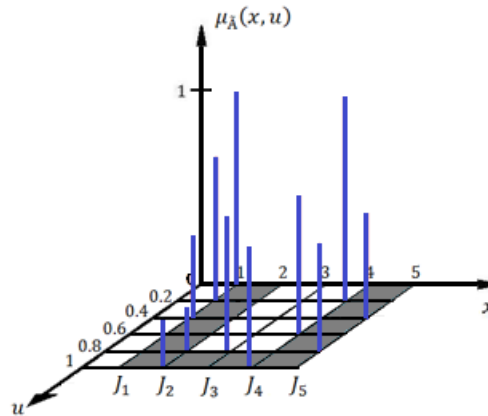
$$\tilde{A} = \left\{ \left((x, u), \mu_{\tilde{A}}(x, u) \right) \mid \forall x \in X, \forall u \in J_x \subseteq [0,1] \right\} \quad (12)$$

Mendel (2017) também definem o conjunto pela união contínua dupla $\int \int$ de todos os termos admissíveis de x e u :

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u) / (x, u) \quad J_x \subseteq [0,1] \quad (13)$$

O termo u representa o valor final da faixa de valores de pertinência J_x possíveis para um único valor real x . A visualização da função de pertinência *fuzzy* tipo-2 é tridimensional e é demonstrada para o caso discreto na Figura 3. Para cada x existem valores finitos de pertinências em u dentro de J_x .

Figura 3 – Função de pertinência discreta *fuzzy* genérica tipo-2



Fonte: Rizol; Mesquita e Saotome (2011)

O conjunto *fuzzy* tipo-2 genérico apresenta a maior flexibilidade e liberdade para modelar incertezas, porém, sua alta complexidade matemática leva a um alto custo de computação e pouca aplicação prática (DOCTOR *et al.*, 2016).

2.1.6.2 Intervalar

Um conjunto *fuzzy* tipo-2 com uma função de pertinência unitária, ou seja, $\mu_{\tilde{A}}(x, u) = 1$, se transforma no conjunto *fuzzy* tipo-2 intervalar (RIZOL; MESQUITA;

SAOTOME, 2011). A nova definição se torna um caso especial da forma genérica de acordo com os mesmos autores:

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} 1/(x, u) J_x \subseteq [0, 1] \quad (14)$$

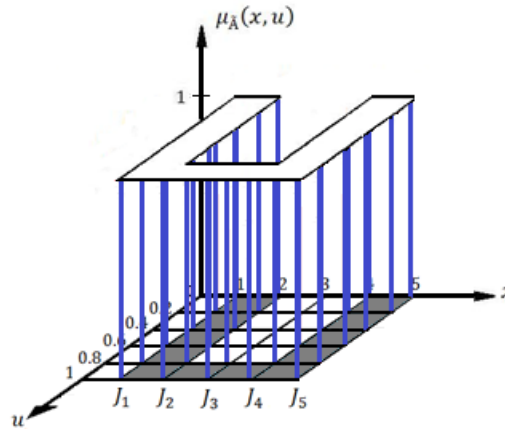
De acordo com Mendel (2017), a definição do conjunto *fuzzy* tipo-2 intervalar pode ser reescrita como:

$$\tilde{A} = \left\{ (x, \mu_{\tilde{A}}(x)) \mid \forall x \in X \right\} \quad (15)$$

$$\tilde{A} = \int_{x \in X} \mu_{\tilde{A}}(x)/x = \int_{x \in X} \left[\int_{u \in J_x} 1/u \right] / x, J_x \subseteq [0, 1] \quad (16)$$

Assim, simplificando sua equação e visualização, demonstrada na Figura 4. Todos os valores de $\mu_{\tilde{A}}(x, u) = 1$, ou seja, não importa o valor de u e a terceira dimensão da visualização não traz informações adicionais com relação à representação bidimensional incluindo o FOU (RIZOL; MESQUITA; SAOTOME, 2011).

Figura 4 – Função de pertinência discreta *fuzzy* tipo-2 intervalar



Fonte: Rizol; Mesquita e Saotome (2011)

A grande vantagem desse tipo de conjunto *fuzzy* tipo-2 é o menor custo computacional, assim a maioria das aplicações reais utilizaram o *fuzzy* tipo-2 intervalar (DOCTOR *et al.*, 2016).

2.2 CONTROLE FUZZY BASEADO EM REGRAS

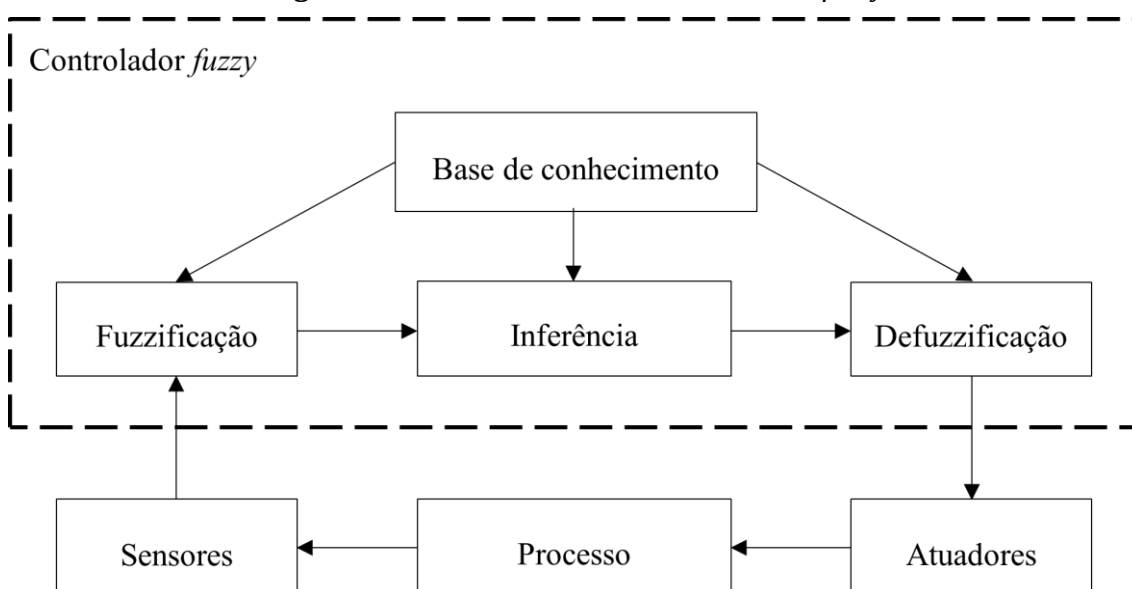
De acordo com Gomide, Gudwin e Tanscheidt (1995), a ideia primária do controle *fuzzy* é se basear no conhecimento de um especialista, como a experiência de um projetista ou operador, da área específica, para tomar decisões que irão controlar o sistema, em vez

de utilizar modelos matemáticos como nos métodos convencionais de controle. Essa abordagem foi motivada de casos específicos em que havia conhecimento disponível de especialistas e a modelagem não era viável praticamente ou economicamente (GOMIDE; GUDWIN; TANSCHIEDT, 1995).

A estrutura básica de um controlador *fuzzy* terá os seguintes componentes ou blocos funcionais: fuzzificação; base de conhecimento; inferência; defuzzificação (SIMÕES; SHAW, 2007). Ainda será conectada ao processo a ser controlado, da maneira mostrada na Figura 5, se baseando em Gomide, Gudwin e Tanscheidt (1995).

Nos próximos tópicos serão especificados cada um dos blocos funcionais da estrutura de um controlador *fuzzy* e a utilização de conjuntos *fuzzy* tipo-2 no controle.

Figura 5 – Estrutura básica do controlador *fuzzy*



Fonte: Adaptada de Gomide; Gudwin e Tanscheidt (1999)

2.2.1 Fuzzificação

Os valores de entrada do controlador, não-*fuzzy*, vêm geralmente de dispositivos de sensoriamento físico ou entradas de computador. Sendo o primeiro passo do ciclo de controle, a fuzzificação irá literalmente fuzzificar as entradas, passar de número para conjunto *fuzzy*, utilizando as funções de pertinências previamente descritas no sistema, ainda é possível converter os valores anteriormente utilizando um fator de escala (SIMÕES; SHAW, 2007).

Para o problema típico de controle, em que o erro $e(t)$ medido entre a referência e o sensor é a verdadeira entrada do controlador, há uma necessidade de converter de um valor de grandeza física, como uma tensão em um potenciômetro, para conjuntos *fuzzy*, como EP (Erro Pequeno), EM (Erro Médio) e EG (Erro Grande), já que são utilizados somente os valores de pertinência no controlador (JÚNIOR; YONEYAMA, 2000).

2.2.2 Base de Conhecimento

A base de conhecimento é composta por outras duas: base de dados e base de regras *fuzzy* (SIMÕES; SHAW, 2007). A primeira contém as funções de pertinência necessárias para as etapas de fuzzificação e defuzzificação, ou seja, elas são previamente construídas, enquanto a última contém as regras lógicas linguísticas que, de certa forma, dá forma aos objetivos e estratégia de controle (SIMÕES; SHAW, 2007).

As regras são premissas lógicas que relacionam as variáveis linguísticas aos seus valores linguísticos e ações correspondentes, regras SE/ENTÃO, com estrutura similar a (SANDRI; CORREA, 1999):

SE (premissa) **ENTÃO** (ação)

Essas são similares a como um ser humano iria controlar algum sistema, utilizando pequenas lógicas com valores linguísticos. Como exemplo, duas regras são criadas para controlar um sistema de ar-condicionado. A variável linguística de entrada será a temperatura do quarto, enquanto a saída será a velocidade do ventilador do sistema:

SE Temperatura é Quente **ENTÃO** Velocidade é Alta.

SE Temperatura é Muito Quente **ENTÃO** Velocidade é Muito Alta.

Múltiplas premissas na mesma regra também podem ser utilizadas com os conectivos *e/ou*. Para m regras que relacionam n variáveis com conetores e , $1 \leq i \leq n$, $1 \leq j \leq m$, sendo que uma variáveis linguísticas de uma regra j é representada por x_i , um valor linguístico de uma regra j é representado por $A_{i,j}$ e a ação e seu respectivo conjunto de uma regra são, respectivamente, y_j e C_j , a estrutura genérica de uma regra R_j com múltiplas premissas e uma ação será (SANDRI; CORREA, 1999):

$R_j : \textbf{SE } x_1 \text{ é } A_{1,j} \textbf{ e } \dots \textbf{ e } x_n \text{ é } A_{n,j} \textbf{ ENTÃO } y_j \text{ é } C_j$

Os valores mostrados no exemplo anterior (Quente, Alta etc.) são conjuntos *fuzzy* relacionados a uma função de pertinência armazenados na base de dados. O próximo passo no controle é a inferência dessas regras, discutida na próxima seção.

2.2.3 Inferência

De acordo com Júnior e Yoneyama (2000), a inferência serve para receber os graus de pertinência dos valores linguísticos associados às variáveis linguísticas de entrada, compará-los com a base de regras e fornecer um ou mais valores linguísticos referentes às variáveis linguísticas de saída.

O processamento final da inferência, ou seja, o fornecimento do conjunto *fuzzy* da saída, é feito em cinco passos (SANDRI; CORREA, 1999), utilizando a estrutura genérica de uma regra R_j da seção 2.2.2 **Base de Conhecimento**:

- a) seja x_i uma variável definida no universo X_i , o valor instantâneo da variável será x_i^* , sendo que $x_i^* \in X_i$;
- b) a compatibilidade de x_i^* com $A_{i,j}$ com a regra R_j será:
 - $\alpha_{i,j} = \mu_{A_{i,j}}(x_i^*)$;
 - $1 \leq i \leq n$;
 - $1 \leq j \leq m$.
- c) para a compatibilidade global da regra com os valores instantâneos x_i^* , é utilizada a t-norma T :
 - $\alpha_j = T(\alpha_{1,j}; \dots; \alpha_{n,j})$;
 - $1 \leq j \leq m$.
- d) o conjunto *fuzzy* C_j da regra é relacionado com a compatibilidade global da regra, para formar um novo conjunto C'_j utilizando um operador de implicação I :
 - $\mu_{C'_j}(y) = I(\alpha_j; \mu_{C_j}(y))$;
 - $\forall y \in Y$.
- e) por último, é feita uma agregação, utilizando o operador ∇ , de cada conjunto *fuzzy* C'_j , ou seja, a contribuição das regras acionadas em um conjunto *fuzzy* final C' :
 - $\mu_{C'}(y) = \nabla(\mu_{C'_1}(y); \dots; \mu_{C'_m}(y))$;
 - $\forall y \in Y$.

Os operadores I , ∇ e a t-norma T são calculados de maneiras diferentes dependendo do modelo utilizado. São demonstradas duas formas de calculá-los utilizando os modelos de Mamdani e Larsen na Tabela 1 (SANDRI; CORREA, 1999).

Tabela 1 – Comparação dos modelos de inferência de controladores *fuzzy*

Modelos	t-norma ($\top$)	Implicação (I)	Agregação (∇)
Mamdani	$\top(a, b) = \min(a, b)$	$I(a, b) = \min(a, b)$	$\nabla(a, b) = \max(a, b)$
Larsen	$\top(a, b) = a * b$	$I(a, b) = a * b$	$\nabla(a, b) = \max(a, b)$

Fonte: Sandri e Correa (1999).

2.2.4 Defuzzificação

De acordo com Simões e Shaw (2007), a defuzzificação transforma o valor linguístico da saída, resultante da inferência das regras, em um valor discreto e numérico que melhor representa o resultado da inferência.

Existem alguns métodos de defuzzificação mais populares, como o centro da área (COA), e outros menos, porém, a escolha de um dependerá das necessidades do projeto, cada um tem suas vantagens e desvantagens, em certos casos podendo ser inadequado a utilização de um método específico (SANDRI; CORREA, 1999).

O método de centro da área se assemelha ao encontrar o centro de gravidade da área formada por conjuntos *fuzzy* resultantes da inferência, calculando o ponto centroide como um valor para a saída final u^* até o atuador. A sua fórmula será o somatório, até o número total de valores linguísticos da saída (N), do valor de posição de centroide de cada área individual (u_i), antes da agregação (∇) no passo final da inferência, multiplicado pela sua pertinência correspondente $\mu_{OUT}(u_i)$, dividido pelo somatório, até o número total de valores linguísticos da saída (N), de cada pertinência $\mu_{OUT}(u_i)$ (SIMÕES; SHAW, 2007):

$$u^* = \frac{\sum_{j=1}^N u_i [\mu_{OUT}(u_i)]}{\sum_{j=1}^N \mu_{OUT}(u_i)} \quad (17)$$

O método é relativamente simples, mas apresenta desvantagens como: maior gasto computacional pela integração numérica e não confiabilidade quando a área final é descontínua.

2.2.5 Controle Fuzzy Tipo-2

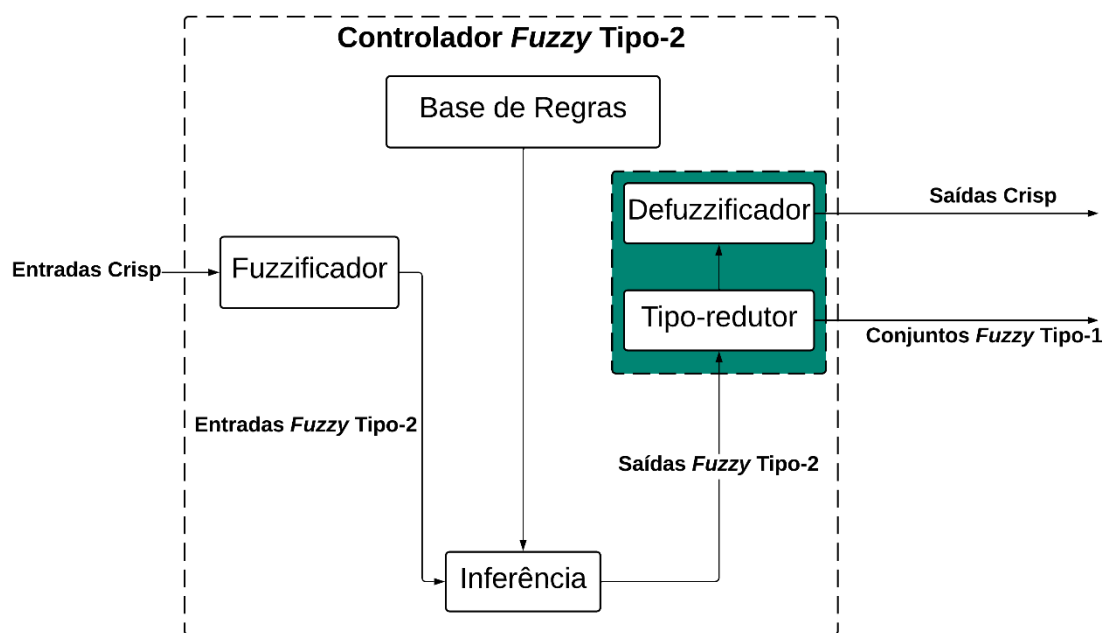
Sistemas de controle que utilizam a lógica *fuzzy* tipo-2 costumam ter vantagens em relação aos de tipo-1, sendo essas (FARIAS, 2019):

- a) conjuntos *fuzzy* tipo-2 utilizam a FOU que permite trabalhar com incertezas na medição da entrada e saída do controlador;
- b) funções de pertinência tipo-2 nas entradas levam a uma maior área de utilização, assim podendo até diminuir o número de regras para um sistema de controle similar tipo-1;
- c) sistemas de controle *fuzzy* tipo-2 são capazes de controlar outros sistemas que não poderiam ser controlados por sistemas tipo-1 utilizando o mesmo número de funções de pertinência.

O controle *fuzzy* tipo-2 é melhor utilizado em aplicações com incerteza na determinação do grau de pertinência, causada por eventuais ruídos na medição, e/ou no modelo matemático não linear e variante no tempo (SOUZA; RIZOL, 2017). Porém, a desvantagem em relação ao tipo-1 é sua maior complexidade computacional dos (FARIAS, 2019).

Para o caso específico de utilizar o tipo intervalar, ou seja, controle *fuzzy* tipo-2 intervalar, a carga computacional é menor, logo, a maioria das aplicações práticas de controle tipo-2 utiliza esse controlador (YAO; WANG, 2017).

A estrutura do controlador *fuzzy* tipo-2 também se difere do tipo-1, sendo a principal diferença a inclusão de mais um passo chamado de tipo-redutor. A Figura 6 demonstra a estrutura de controlador *fuzzy* tipo-2.

Figura 6 – Estrutura básica do controlador *fuzzy* tipo-2

Fonte: Adaptada de Souza e Rizol (2017)

A figura anterior mostra os blocos funcionais de um controlador *fuzzy* tipo-2. Comparando com a estrutura do controlador tipo-1 (Figura 5), o bloco Tipo-redutor reduz a ordem do sistema *fuzzy*, nesse caso de tipo-2 para tipo-1, enquanto o bloco Defuzzificador, também presente no controlador tipo-1, transforma de volta para um valor numérico ou *crisp*. A seguir, será descrito sobre o Tipo-redutor assim como será citado alguns algoritmos clássicos utilizados.

2.2.5.1 Tipo-redutor

De acordo com Mendel (2017), um sistema *fuzzy* tipo-2 tem dois caminhos possíveis em transformar um conjunto *fuzzy* tipo-2 para um valor numérico:

- Defuzzificação direta:** Transformação de um conjunto *fuzzy* tipo-2 para um número diretamente;
- Tipo-redutor + Defuzzificação:** Primeiro o conjunto é transformado em tipo-1 pelo tipo-redutor e por último acontece a defuzzificação do conjunto *fuzzy* tipo-1.

Historicamente o nome tipo-redutor foi utilizado para referir-se a um algoritmo que transforma do tipo-2 para o tipo-1, ainda assim alguns autores classificam a

defuzzificação direta como tipo-redutor porque transforma do tipo-2 para tipo-0, ou seja, o valor numérico em si (MENDEL, 2017).

Vários algoritmos de Tipo-redutor foram criados e são propostos até recentemente, como visto em Chen *et al.* (2019). Os mais populares são os algoritmos de Karnik-Mendel (KM), Karnik-Mendel melhorado (EKM) e Algoritmo Iterativo com Condição de Parada melhorado (EIASC). Todos são iterativos e são baseados na extensão do princípio matemático dos métodos de defuzzificação para controladores tipo-1, uma média ponderada estendida para intervalos em controladores tipo-2 intervalares (IWA) (MENDEL, 2017).

2.2.6 Controladores *Fuzzy*

Em Simões e Shaw (2007), dois modelos de controladores *fuzzy* são descritos: Mamdani ou baseado em regras, o primeiro a ser introduzido, em 1975 e o Tagaki-Sugeno (TS) ou paramétricos em 1985. Os dois se diferem justamente nas regras utilizadas e o processo de defuzzificação, em que Mamdani utiliza regras com premissas e ações que são conjuntos *fuzzy* e defuzzificação como foi explicado nas seções 2.2.2 **Base de Conhecimento** a 2.2.4 **Defuzzificação** deste capítulo. O modelo TS inclui regras com premissas como conjuntos *fuzzy* e equações matemáticas para as ações, assim dispensando a defuzzificação normal e melhorando sua eficiência computacional (VASCONCELLOS, 2017).

Neste trabalho, foi descrito e será o utilizado o modelo Mamdani, ou controle *fuzzy* baseado em regras de acordo com Simões e Shaw (2007). Tanto controladores *fuzzy* tipo-1 (T1.FLC) quanto tipo-2 intervalares (IT2.FLC) serão utilizados no controle da anestesia. A seguir, a próxima seção irá tratar das principais características da Anestesia geral e seu controle.

2.3 ANESTESIA GERAL

O procedimento anestésico é ideal em condições cirúrgicas tanto para o cirurgião quanto para o paciente porque o permite não sentir dor ou se mover durante a cirurgia (KHODAEI *et al.*, 2019).

Taheriyani, Ghafourian e Noori (2018) afirmam que, durante o procedimento cirúrgico, o paciente precisa estar sob efeito da anestesia, ou seja, sem sensibilidade a dor ou qualquer outro estímulo.

Nas seguintes seções será discutida a definição da anestesia geral, os procedimentos utilizados por médicos anestesistas, os erros de dosagem e possíveis consequências na anestesia geral, os efeitos de medicamentos anestésicos no corpo humano, considerando esse como um sistema dinâmico altamente não linear, e os métodos de controle automático do processo de anestesia encontrados em pesquisas recentes.

2.3.1 Definição da Anestesia

A anestesia tem sua definição associada com a perda de consciência e falta de resposta a um estímulo nocivo (KHODAEI *et al.*, 2019).

Sadati, Hosseinzadeh e Dumont (2018) definem a anestesia como um estado temporário reversível de inconsciência, ausência de percepção da dor e relaxamento muscular, ainda citando três componentes em que consiste o procedimento anestésico, sendo eles: inconsciência, analgesia e bloqueio neuromuscular.

2.3.2 Procedimentos e Sinais Biomédicos Utilizados

O médico anestesista tem à sua disposição o conhecimento de vários procedimentos específicos, divididos em partes ou fases da anestesia, e medições de sinais biomédicos para auxiliá-lo em seu trabalho.

As três principais fases de anestesia são: indução, manutenção e recuperação. A primeira serve para induzir o fármaco no paciente até sua inconsciência antes do começo da cirurgia, enquanto a próxima mantém a dosagem ao decorrer do procedimento. Por último, a fase de recuperação tem como objetivo parar a infusão do remédio até a retomada de consciência do paciente (KHODAEI *et al.*, 2019).

De acordo com Doctor *et al.* (2016), a principal responsabilidade do médico anestesista é manter o relaxamento muscular, inconsciência e analgesia utilizando medicamentos. Vários sinais são utilizados na prática para verificar o estado de anestesia do paciente, como a atividade elétrica nas células musculares, utilizando a eletromiografia (EMG, do inglês *Electromyography*), e no cérebro, utilizando a eletroencefalografia

(EEG, do inglês *Electroencephalography*), além da pressão sanguínea (BP, do inglês *Blood Pressure*), entre outros, porém, a pressão sanguínea é o sinal mais confiável para medir o nível de consciência e profundidade de anestesia (DoA, do inglês *Depth of Anesthesia*).

2.3.3 Erro de Dosagem na Anestesia

Errar desempenhando funções complexas é humano, mas pode causar graves consequências. Na área de anestesiologia encontra-se profissionais que planejam, manuseiam e administram muito frequentemente remédios sem a consulta de outros profissionais médicos, assim é criado um cenário favorável a erros humanos (ERDMANN *et al.*, 2016).

Em um trabalho feito por Dhawan *et al.* (2017), os tipos de erros de medicação que podem ocorrer durante o procedimento anestésico são erros durante: a preparação do medicamento, a administração do fármaco e registro do medicamento administrado.

Dentro esses tipos, os erros específicos são bem variados e suas causas são encontradas nas seguintes categorias:

- a) atos inseguros: relacionadas ao próprio anestesiológico, como violações de normas ou deslizes;
- b) fatores do local de trabalho: relacionadas ao ambiente hospitalar, como políticas internas, interrupções ou a própria cultura do ambiente de trabalho;
- c) decisões organizacionais: relacionadas à administração, como decisões de alto nível.

A maioria dos erros de anestesia ocorrem no período de manutenção, enquanto os principais erros de administração são de comissão, quando uma dose ou medicamento no tempo ou local não foram corretamente induzidos no paciente, e omissão, quando o fármaco não é administrado no tempo correto ou não é registrado corretamente (ANNIE; THIRILOGASUNDARY; KUMAR, 2019). Um dos tipos específicos de erros de administração é o erro de dosagem do medicamento, que foi considerado o erro com a segunda maior incidência no estudo realizado por Erdmann *et al.* (2016).

As consequências de qualquer erro de medicamento podem chegar à morte. Levando em consideração uma classificação (DHAWAN *et al.*, 2017), a morte de um paciente está localizada na última categoria com a letra I, sendo que a categoria A indica

a ausência de um erro relevante. As outras categorias dividem a ocorrência de um erro sem danos (categoria B até D) e com danos temporários ou permanentes (categoria E até H) ao paciente. Para erros de administração de medicamentos da anestesia, overdose é mostrado como o segundo erro de maior incidência (38%) de acordo com a pesquisa do autor citado anteriormente, depois do erro de medicação (48%), enquanto os outros são administração incorreta (8%), subdosagem (4%) e omissão (2%). Isso significa que o erro de dosagem, principalmente a superdosagem, é relevante. Suas consequências são maior tempo de recuperação da cirurgia em caso de sobredose e até acordar no meio do procedimento se houver subdosagem (DOCTOR *et al.*, 2016).

Duas pesquisas recentes se concentraram em verificar a incidência de erros de administração de fármacos na anestesia regionalmente e nacionalmente. A primeira em Erdmann *et al.* (2016) se concentrou tanto na quantidade quanto na especificidade do erro em médicos anesthesiologists de Santa Catarina, com um total de 61 que responderam ao questionário. Os resultados mostraram que 91,8% dos médicos haviam cometido algum erro, sendo erros de substituição do medicamento (68,4%), erro de dosagem (49,1%) e omissão (35%) os tipos mais comuns. As causas mais comuns para cometer o erro foram distração e fadiga (64,9%) e a leitura errada dos rótulos de embolas ou seringas (54,4%). Enquanto a outra pesquisa mais recente em Annie, Thirilogasundary e Kumar (2019) levou em consideração todos os médicos pertencentes à sociedade indiana de anesthesiologists, com um total de 978 respostas ao questionário, que resultou em 75,6% admitindo algum erro. Dessa forma, erros de administração e dosagem são comuns, logo a tecnologia pode ajudar a controlar sua incidência.

2.3.4 Medicamentos Utilizados na Anestesia

Existem vários tipos de medicamentos na anestesia. Doctor *et al.* (2016) cita Atracúrio e Cis-Atracúrio como medicamentos intravenosos somente para relaxamento muscular e Isoflurano e Sevoflurano como medicamentos inalatórios somente para inconsciência, esses sendo chamados de anestésicos.

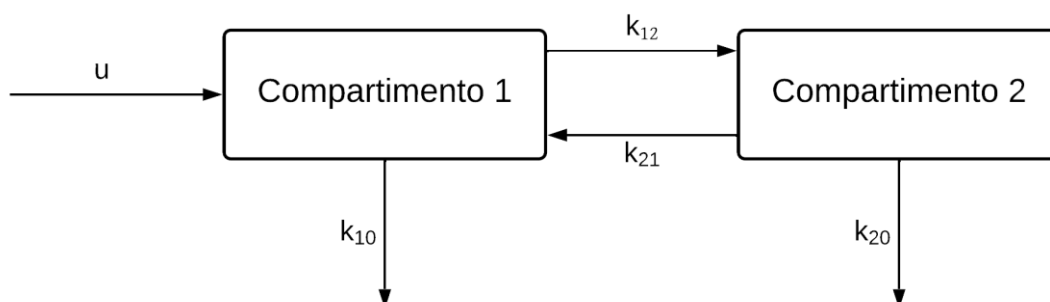
Nas seguintes seções será discutido sobre os conceitos de farmacocinética e farmacodinâmica de medicamentos utilizados na anestesia, o modelo farmacocinético farmacodinâmico (PK/PD) completo e o comportamento de dois medicamentos pertinentes ao trabalho.

2.3.4.1 Farmacocinética

O conceito de farmacocinética de um fármaco é sua concentração no sangue seguindo uma função que varia no tempo e depende da concentração introduzida no corpo (DOCTOR *et al.*, 2016).

O modelo de compartimentos (COPOT, 2020) é utilizado para ilustrar a ideia, demonstrado na Figura 7.

Figura 7 – Modelo Farmacocinético com dois compartimentos



Fonte: Adaptada de Shieh *et al.* (2009)

A Figura 7 mostra o modelo de dois compartimentos farmacocinético com a entrada do fármaco representada pela variável u , parâmetros k_{10} e k_{20} como parâmetros de eliminação do remédio e k_{12} e k_{21} são constantes de movimento da concentração entre compartimentos (SHIEH *et al.*, 2009). O compartimento 1 representa o plasma sanguíneo (parte líquida do sangue) e o compartimento 2 representa qualquer outro compartimento, como músculo ou gordura (COPOT, 2020).

Por último, a Equação 18 a seguir descreve a concentração do fármaco no plasma sanguíneo x_1 para o modelo de dois compartimentos descrito em Copot (2020) considerando que o compartimento 1 tem o volume v_1 :

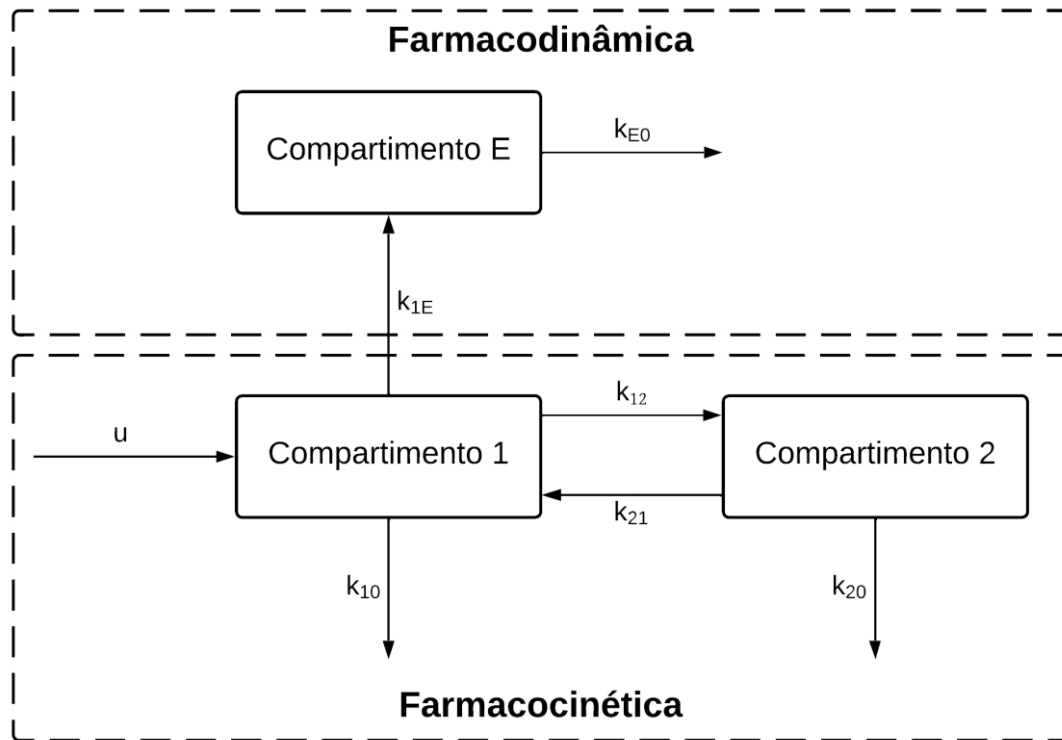
$$\dot{x}_1 = -(k_{10} + k_{12})x_1 + k_{12}x_2 + \frac{u}{v_1} \quad (18)$$

2.3.4.2 Farmacodinâmica

Farmacodinâmica se concentra na relação entre o efeito desejado a partir da concentração do fármaco em uma função dinâmica (DOCTOR *et al.*, 2016). As duas funções são necessárias para entender como o remédio funciona no corpo humano e criam

a estrutura farmacocinética farmacodinâmica (PK/PD, do inglês *Pharmacokinetics* / *Pharmacodynamics*). Essa também é explicada pelo modelo de compartimentos, incluindo o compartimento do efeito (E) demonstrado na Figura 8.

Figura 8 – Modelo PK/PD com dois compartimentos



Fonte: Adaptada de Shieh *et al.* (2009)

A figura acima mostra as delimitações de farmacocinética e farmacodinâmica, formando o modelo PK/PD que inclui o compartimento E, representando a “concentração” do efeito do remédio x_E em uma função de sua concentração no plasma sanguíneo x_1 . Realisticamente o volume do compartimento E será muito menor em relação ao compartimento principal ($v_E \ll v_1$), assim $k_{1E} \approx 0$ (COPOT, 2020).

Parte da equação da farmacodinâmica será similar à farmacocinética, ou seja, a relação da variação da concentração do Efeito $\dot{x}_E$ e a concentração do compartimento principal x_1 , porém, só descreve a parte linear, enquanto isso, a parte não linear é descrita pela **Equação de Hill** (Khalifa *et al.*, 2020):

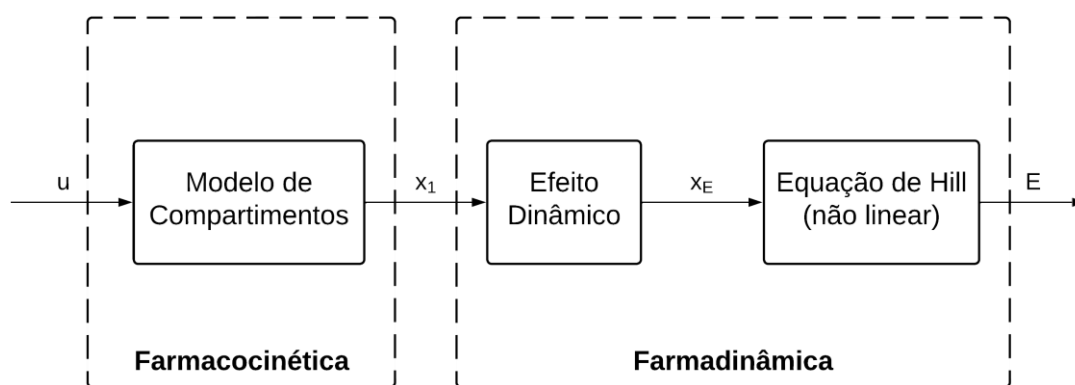
$$E = \frac{E_{max}}{1 + \frac{x_E(50)^\alpha}{x_E^\alpha}} \quad (19)$$

O efeito real E é descrito como um intervalo de valores, tendo seu máximo valor E_{max} , valor quando tem 50% de concentração $x_E(50)$ e uma constante α para a curva sigmoide. Para a aplicação de controle também é considerado que o valor mínimo do efeito seja nulo ($E_0 = 0$) e o valor máximo seja um ($E_{max} = 1$) (COPOT, 2020).

2.3.4.3 Modelo PK/PD

O modelo PK/PD utiliza das equações dinâmicas de cada uma de suas partes em série. A Figura 9 mostra o de maneira simplificada toda a estrutura em diagrama de blocos.

Figura 9 – Modelo PK/PD em diagrama de blocos



Fonte: Adaptada de Copot (2020)

O diagrama de blocos da figura anterior mostra o comportamento em série para o cálculo do efeito final. Cada bloco é uma equação dinâmica com capacidade de representação em função de transferência ou espaço de estados para os dois primeiros, incluindo possíveis atrasos de transporte para o efeito dinâmico (COPOT, 2020). A seguir, serão mostrados dois exemplos de modelagem PK/PD de fármacos que serão utilizados neste trabalho.

2.3.4.4 Atracúrio

Esse fármaco é utilizado para relaxamento muscular ou bloqueio neuromuscular na anestesia de modo intravenoso (Khalifa *et al.*, 2020).

Seu modelo PK/PD pode ser representado em duas partes (DOCTOR *et al.*, 2016). A primeira é uma função de transferência da farmacocinética e farmacodinâmica linear com a unidade em minutos, atraso de transporte de 1 minuto e infusão do atracúrio $U_{(inf)}(s)$:

$$G_{PK-PD(atr)}(s) = \frac{X_E(s)}{U_{(inf)}(s)} = \frac{(1 + 10,64s)e^{-s}}{(1 + 4,81s)(1 + 34,42s)(1 + 3,08s)} \quad (20)$$

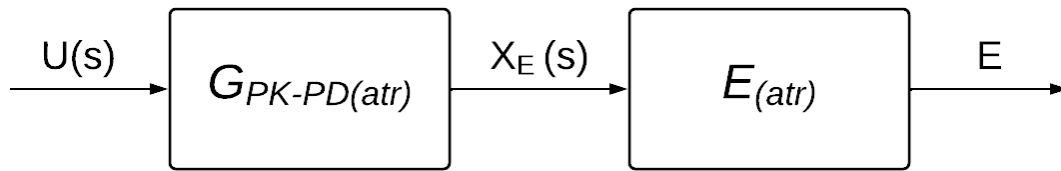
Esses parâmetros são representativos de estudos anteriores citados na referência, demonstrando valores comuns em uma função de transferência de terceira ordem, incluindo o atraso de transporte de 1 minuto, ou seja, demora 1 minuto para a saída responder a algum estímulo da entrada.

Já a segunda parte representa o comportamento não linear utilizando a equação de Hill com os seguintes parâmetros:

$$E_{(atr)} = \frac{X_E^{2,98}}{X_E^{2,98} + (0,404)^{2,98}} \quad (21)$$

O modelo completo em diagrama de blocos será uma combinação das duas equações em série (Figura 10).

Figura 10 – Modelo PK/PD do Atracúrio em diagrama de blocos



Fonte: Adaptada de Copot (2020)

Por último, o efeito E de relaxamento muscular será normalizado de 0 a 1 (100%), ou seja, indica quanto o paciente está imobilizado.

2.3.4.5 Isoflurano

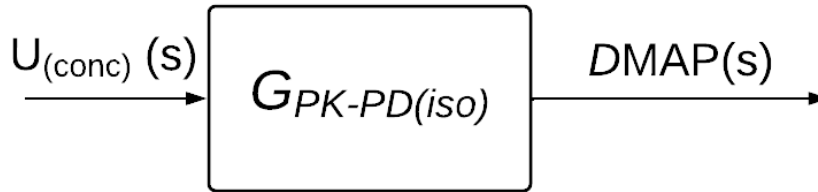
O isoflurano é um remédio inalatório utilizado na anestesia para incidir a inconsciência ou profundidade da anestesia (DoA) pela diminuição da pressão arterial média (MAP) (TAHERIYAN; GHAFOURIAN; NOORI, 2018).

Na Figura 11 está o diagrama de blocos da modelagem PK/PD do isoflurano. A modelagem PK/PD é pode ser demonstrada por uma equação dinâmica linear, sem a

necessidade da equação de Hill porque o fármaco é inalado, com a unidade em minutos, atraso de transporte de 0,42 minuto, concentração do isoflurano $U_{(conc)}(s)$ e efeito de diminuição da pressão arterial média $\Delta MAP(s)$ (DOCTOR *et al.*, 2016):

$$G_{PK-PD(iso)}(s) = \frac{\Delta MAP(s)}{U_{(conc)}(s)} = \frac{-15e^{-0,42s}}{1 + 2s} \quad (22)$$

Figura 11 – Modelo PK/PD do Isoflurano em diagrama de blocos



Fonte: Adaptada de Copot (2020)

2.3.4.6 Modelo Multivariável do Corpo Humano para Anestesia

O procedimento de anestesia consiste em três componentes principais já citados anteriormente na seção 2.3.1 **Definição da Anestesia** deste capítulo. De todos eles, a analgesia está mais ligada a condições pós-operatórias, logo não é necessária para a modelagem e controle automática da anestesia (TAHERIYAN; GHAFOURIAN; NOORI, 2018). Os outros dois podem ser modelados utilizando os modelos PK/PD dos fármacos descritos nas seções 2.3.4.4 Atracúrio e 2.3.4.5 Isoflurano, dessa forma é possível criar um modelo do corpo humano no procedimento anestésico.

Seguindo a modelagem em Shieh *et al.* (2009), as equações formam uma matriz multivariável de funções de transferência, tendo as entradas $U_1(s)$ como a infusão do atracúrio e $U_2(s)$ como a concentração do isoflurano. As saídas são Relaxamento Muscular $RM(s)$, efeito do atracúrio, e Pressão Sanguínea $PS(s)$, efeito do isoflurano. O termo Pressão Arterial Média (MAP, do inglês *Mean Arterial Pressure*) será definido como o mesmo de $PS(s)$ neste trabalho para o controle. As equações seguirão a nomenclatura comum do controle multivariável, por exemplo, $G_{11}(s)$ será a função de transferência da entrada 1 $U_1(s)$ para a saída 1 $RM(s)$:

$$G_{11}(s) = \frac{X_E(s)}{U_1(s)} = G_{PK-PD(atr)}(s) = \frac{(1 + 10,64s)e^{-s}}{(1 + 4,81s)(1 + 34,42s)(1 + 3,08s)} \quad (23)$$

$$G_{22}(s) = \frac{PS(s)}{U_2(s)} = G_{PK-PD(iso)}(s) = \frac{-15e^{-0,42s}}{1 + 2s} \quad (24)$$

Ainda serão adicionadas interações entre um remédio para o outro efeito. O atracúrio tem efeitos mínimos na diminuição MAP, pode ser desconsiderado, mas o isoflurano $U_2(s)$ tem o seguinte efeito PK/PD no relaxamento muscular (DOCTOR *et al.*, 2016):

$$G_{12}(s) = \frac{0,27e^{-s}}{(1 + 2,83s)(1 + 1,25s)} \quad (25)$$

A modelagem linear completa pode ser descrita na forma de matriz de funções de transferência:

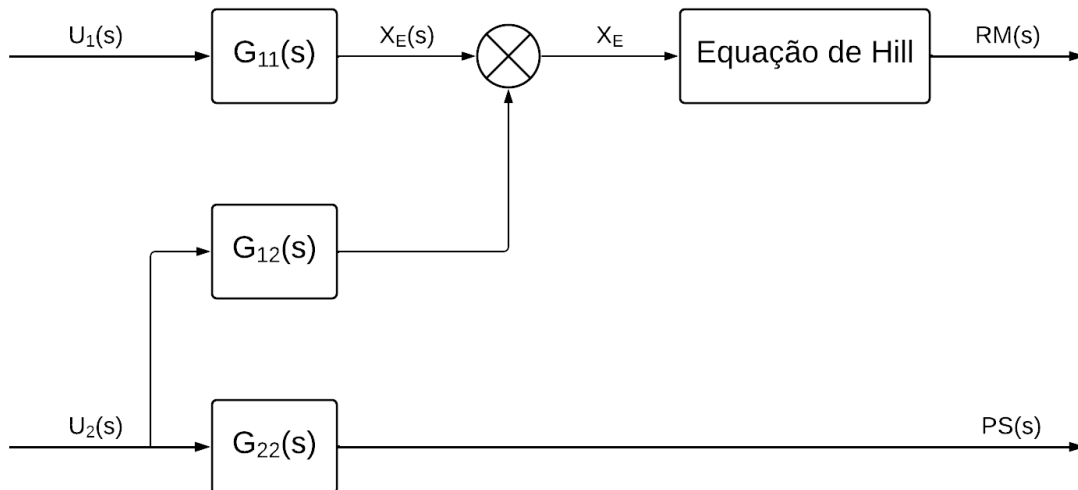
$$\begin{bmatrix} RM(s) \\ PS(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ 0 & G_{22}(s) \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} \quad (26)$$

Por último, o comportamento não linear do efeito causado principalmente pelo atracúrio pode ser modelado pela Equação de Hill descrita anteriormente na seção 2.3.4.4 Atracúrio:

$$RM = \frac{X_E^{2,98}}{X_E^{2,98} + (0,404)^{2,98}} \quad (27)$$

Dessa forma, juntando a modelagem de matriz de funções de transferência e o efeito não linear, obtém-se o modelo completo multivariável não linear. A Figura 12 demonstra melhor a ordem das equações.

Figura 12 – Modelo PK/PD do corpo humano na anestesia em diagrama de blocos



Fonte: Adaptada de Taheryian; Ghafourian e Noori (2018)

A figura anterior mostra como as equações devem ser utilizadas para encontrar o verdadeiro efeito dos fármacos no corpo humano. Esse modelo servirá como auxílio na simulação do controle da anestesia geral, tópico da próxima seção.

Todos esses valores específicos do modelo são baseados em algum tipo de pesquisa para encontrar tais valores, mas na realidade eles podem variar de paciente para paciente (TAHERYIAN; GHAFOURIAN; NOORI, 2018). De acordo com Doctor *et al.* (2016), os parâmetros do modelo podem variar tanto entre pacientes diferentes (inter paciente), quanto no mesmo paciente ao decorrer da cirurgia (intra paciente). Portanto, o modelo tem suas limitações, mas consegue mostrar várias dificuldades com relação ao controle da anestesia geral.

2.3.5 Controle da Anestesia Geral

As primeiras pesquisas relacionadas ao controle e automação da anestesia não são recentes ou até mesmo deste século. De acordo com Alexander, Romito e Çobanoğlu (2020), Bickford criou o primeiro sistema de controle de malha fechada para a manutenção da anestesia geral em 1950 utilizando sinais de eletroencefalograma (EEG) como sensoriamento.

Mais recentemente, o uso de técnicas avançadas de controle automático inteligente, dentre eles o controle *fuzzy*, e não linear para anestesia se mostrou bem mais adequado em atender às necessidades inerentes que dificultam esse problema, como o comportamento dinâmico não linear de fármacos (farmacodinâmica) e variabilidade de parâmetros do paciente (TAHERYIAN; GHAFOURIAN; NOORI, 2018).

As próximas seções irão descrever sobre o controle *fuzzy* da anestesia, além de outras abordagens mais avançadas, e a proposta do trabalho.

2.3.5.1 Controle *Fuzzy* da Anestesia

O trabalho de Shieh *et al.* (2009) está na literatura como um exemplo da utilização de um controle de anestesia com lógica *fuzzy*, nesse ele demonstra a robustez de um controlador de lógica *fuzzy* autoajustável (SOFLC) com simulações em um problema de controle multivariável (MIMO), além de um novo método de decomposição e, consequentemente, simplificação desse controlador. Tanto Doctor *et al.* (2016) quanto Wei *et al.* (2020) experimentam com controladores *fuzzy* tipo-2 (T2FLC) e comparam

seus desempenhos robustos entre duas ou mais versões em simulações, sendo que o primeiro compara a resposta com o controlador *fuzzy* tipo-1 (T1FLC) em versão SOFLC e o último utiliza programação genética (GP) para otimizar seus parâmetros do controlador, assim tendo várias versões do mesmo. Nas pesquisas de dois autores mencionados anteriormente (SHIEH *et al.*, 2009; Doctor *et al.*, 2016), são utilizadas metodologias similares, ou seja, enfrentam o problema de controle multivariável da anestesia, controlando as variáveis: Relaxamento Muscular e Pressão Sanguínea e utilizam os medicamentos: Atracúrio e Isoflurano como variáveis manipuladas do sistema e seus resultados são dados por simulações com modelos matemáticos alimentados com dados reais que preveem o comportamento dinâmico de vários pacientes ao serem introduzidos os fármacos no corpo humano. O trabalho mais recente (WEI *et al.*, 2020) controla o Índice Bispectral (BIS) e consequentemente a profundidade de anestesia (DoA) em simulação utilizando o fármaco Propofol.

Algumas pesquisas validaram os controladores com estudos clínicos em pacientes reais além da simulação, como em Méndez *et al.* (2016) e Méndez *et al.* (2018). No primeiro foi utilizado um controlador *fuzzy* adaptativo e preditivo, utilizando modelo de controle preditivo (MPC) e algoritmos genéticos (GA), para o controle do BIS e DoA em 16 pacientes com bons resultados em comparação a estudos anteriores similares. O último se concentrou em automatizar o processo anestésico utilizando o controlador *fuzzy* e um sistema supervisor para trabalhar junto ao anestesista. O experimento teve 81 pacientes divididos em dois grupos, um utilizando o sistema automatizado e outro com técnicas de dosagem manual, e conseguiu melhores resultados com o primeiro em acurácia e mantimento da anestesia.

2.3.5.2 Outros Tipos de Controle da Anestesia

Outra ferramenta efetiva em controle inteligente é a rede neural artificial (RNA), que pode ser utilizada de maneira híbrida com a lógica *fuzzy*, assim tendo o controle neuro-*fuzzy*. Esse tipo de controle foi utilizado na pesquisa de Khodaei *et. al* (2019) para uma administração intravenosa de anestesia.

Pela natureza não linear do controle da anestesia, uma abordagem é utilizar métodos de controle não linear como em Ilyas *et al.* (2017) que utilizou o método de controle por métodos deslizantes (SMC – *Sliding Mode Control*) para a administração de propofol, controlando assim a inconsciência do paciente.

2.3.5.3 Proposta do Trabalho

Nos procedimentos de controle de regulagem de anestesia.

- i. se for utilizado o medicamento Atracúrio para controlar o relaxamento muscular;
- ii. se for utilizado o medicamento Isoflurano para controlar a pressão sanguínea;
- iii. se forem utilizados conjuntos *fuzzy* tipo-2 de intervalo baseados em dados biomédicos.

Então, controle *fuzzy* tipo-2 é uma alternativa para resolver a técnica de aplicação de anestesia.

3 MATERIAIS E MÉTODOS

Nesse capítulo serão especificados os materiais e métodos utilizados no trabalho, além da metodologia da pesquisa feita. Os projetos de controladores *fuzzy* tipo-1 e tipo-2 também são demonstrados. Será evidenciado que o controlador *fuzzy* não é necessariamente físico mas pode ser implementado fisicamente em um computador por meio de um algoritmo.

3.1 MATERIAIS UTILIZADOS

Grande parte do trabalho foi concluído por meio de programas de computador utilizados para projetar, implementar e avaliar dispositivos de controle e automação, portanto, os materiais utilizados estão divididos em: *software* e *hardware*, descritos no Quadro 1.

Para o desenvolvimento deste trabalho foi utilizado ambiente virtual para demonstrar a proposta do trabalho, conforme tabela a seguir:

Quadro 1 – Equipamentos utilizados no trabalho

<i>Software</i>	<i>Hardware</i>
Linguagem C	Computador pessoal com processador Intel® Core™ i7-8565U, memória RAM DDR4 de 20 GB
Ambiente MATLAB: versão 9.11 (2021b)	Arduino UNO R3
Fuzzy Logic Toolbox: versão 2.8.2 (2021b)	Mini Motor DC 3V a 6V
Control System Toolbox: versão 10.11 (R2021b)	Mini Bomba Submersível DC 3V – 6V 120L
Simulink: versão 10.4 (2021b)	Módulo Sensor de Luz LDR
Funções MATLAB	Sensor Ultrassônico HC-SR04
MATLAB Support Package for Arduino Hardware: versão 21.2.0	Protoboard 2420 furos Minipa MP-240A
	Cabos jumper
	Ferro de soldar Hikari Power 60
	Fonte com 4 pilhas AA (4 x 1,5V = 6V)

Fonte: Elaborado pelo Autor (2021)

O Quadro 1 lista todos os materiais utilizados para a realização desse trabalho separadas entre componentes físicos computacionais, eletrônicos e acessórios,

denominados *hardware*, e componentes digitais utilizados pelo computador, classificados como *software*. A seguir será demonstrada a metodologia da pesquisa realizada.

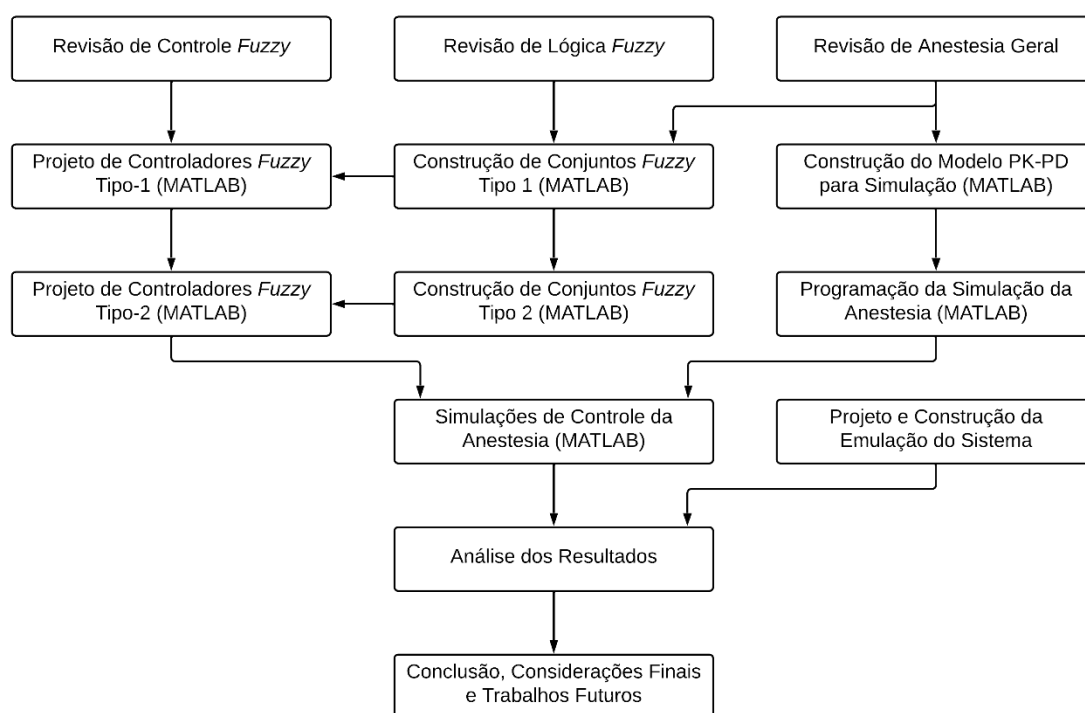
3.2 METODOLOGIA DO TRABALHO

Segundo Silveira e Córdova (2009), a metodologia da pesquisa considera diversos fatores: quanto à abordagem, pode ser qualitativa ou quantitativa, e pode ser dividida quanto à natureza, sendo básica (puramente teórica) ou aplicada (aplicação prática imediata). Ainda é possível dividir a metodologia quanto aos seus objetivos, como pretende lidar com os fenômenos inerentes da pesquisa, e quanto aos seus procedimentos, quais ferramentas serão utilizadas para atingir o objetivo de responder o questionamento.

Nesse contexto, quanto à abordagem, a metodologia deste trabalho apresenta as características descritas a seguir:

- a) abordagem quali-quantitativa – porque ele faz uma interpretação e análise dos resultados numéricos das técnicas de controle *fuzzy* pesquisadas contidas no texto do trabalho, e irá trabalhar com dados numéricos de simulação e dados referenciais médicos de relaxamento muscular e pressão sanguínea média e desvio padrão de 15 pacientes de acordo com Doctor *et al.* (2016) para compor os conjuntos *fuzzy* tipo-2 intervalares utilizados no trabalho;
- b) pesquisa aplicada – busca melhorar a eficiência das técnicas modernas de anestesia;
- c) pesquisa explicativa – irá confirmar ou não as hipóteses levantadas no trabalho com relação ao controle automático de dosagem anestésica a partir dos dados coletados por meio de simulações computacionais do controlador *fuzzy*;
- d) pesquisa bibliográfica – irá utilizar dados médicos de pacientes e modelos matemáticos já existentes na bibliografia para realizar simulações do algoritmo de controle *fuzzy*; e
- e) estudo de caso – busca apresentar um estudo definido sobre o uso da técnica de controle *fuzzy* como meio para a automação na anestesiologia.

Segundo as técnicas metodológicas citadas anteriormente, a seguir será descrito através de um fluxograma a proposta do trabalho mostrado na Figura 13.

Figura 13 – Proposta do trabalho em fluxograma

Fonte: Elaborada pelo Autor (2021)

Na Figura 13, o fluxograma mostra vários níveis de processos em paralelo que foram pré-requisitos para o próximo nível. Primeiramente, uma revisão bibliográfica extensa sobre os assuntos pertinentes em Controle da Anestesia, esses sendo Controle *Fuzzy*, Lógica *Fuzzy* e Anestesia geral, foi realizada. Depois da revisão, o MATLAB foi utilizado para projetar os controladores *fuzzy* a partir dos conjuntos *fuzzy* tipo-1 e tipo-2. Cada conhecimento acumulado sobre lógica e controle *fuzzy* foi necessário para construir e simular, entretanto, os conjuntos *fuzzy* também precisaram do conhecimento da anestesia geral. Em paralelo, foi concluída a modelagem e simulação da anestesia pelo modelo PK/PD no ambiente MATLAB. A simulação final do Controle da Anestesia necessitou do modelo e controlador para gerar os resultados finais, enquanto isso, foi projetado e construído a emulação ou representação física do sistema. A análise final dos resultados do trabalho foi concluída após a parte experimental como pré-requisitos para a conclusão e considerações finais do trabalho, além de trabalhos futuros.

Para ser concretizado o trabalho, os projetos de controladores a seguir serão descritos em detalhe.

3.3 PROJETO DOS CONTROLADORES FUZZY TIPO-1 (T1.FLC)

De maneira geral, o controlador *fuzzy* moderno é baseado em um algoritmo de computador (ABDO-ALLAH, 2018), diferente de outros controladores que podem ser tanto algoritmos quanto circuitos elétricos. Portanto, será necessária uma plataforma para desenvolver tal código.

A ferramenta de programação e cálculos numéricos MATLAB (*MATrix LABoratory*) foi utilizada para projetar os controladores, ou seja, definir todas as partes dos sistemas. Essa ferramenta contém vários pacotes de programação, chamados de *toolbox*, contendo comandos específicos ou interfaces que auxiliam na resolução de problemas computacionais. No MATLAB existe um *toolbox* proprietário para projetar controladores *fuzzy*, chamado de *Fuzzy Logic toolbox*, que foi utilizado para projetar e avaliar as respostas da performance do controlador *fuzzy*.

De acordo com os resultados de Shieh *et al.* (2009), o controlador *fuzzy* se beneficiou com a inclusão de mais duas entradas para a diminuição do erro estacionário, as integrais dos erros do relaxamento muscular e pressão sanguínea, por isso foram projetados dois tipos de controladores *fuzzy* tipo-1 descritos nas próximas subseções.

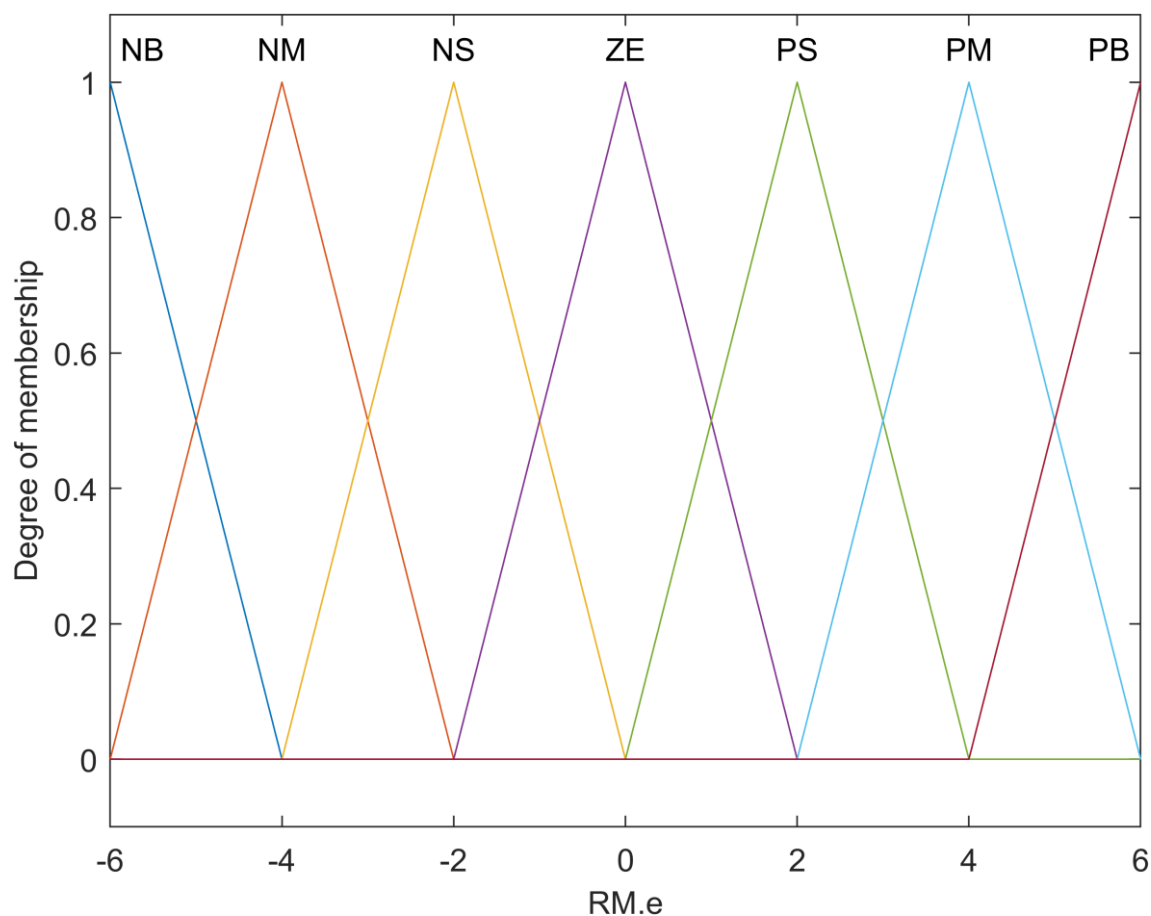
Um método para decompor sistemas *fuzzy* multivariáveis baseado em regras é utilizado para simplificar a estrutura dos controladores *fuzzy* de duas ou quatro entradas e duas saídas em múltiplos sub-controladores de duas entradas e uma saída, assim simplificando as bases de regras (DOCTOR *et al.*, 2016).

Nas seções seguintes, serão descritas as especificações de projeto dos controladores *fuzzy* tipo-1 e dois tipos de controladores projetados, um deles com duas entradas e outro com quatro entradas, incluindo diagramas funcional e de decomposição.

3.3.1 Especificações de projeto dos controladores T1.FLC

Os controladores T1.FLC serão projetados de acordo com a estrutura mostrada na Figura 5. Para cada bloco será feita uma escolha de que tipo e método utilizar.

Primeiramente, os blocos de Fuzzificação e Defuzzificação utilizarão conjuntos *fuzzy* triangulares com, respectivamente, sete e quatro valores linguísticos para cada entrada e cada saída como demonstrado na Figura 14 e Figura 15.

Figura 14 – Valores linguísticos das entradas dos controladores T1.FLC

Fonte: Adaptada de Shieh *et al.* (2009)

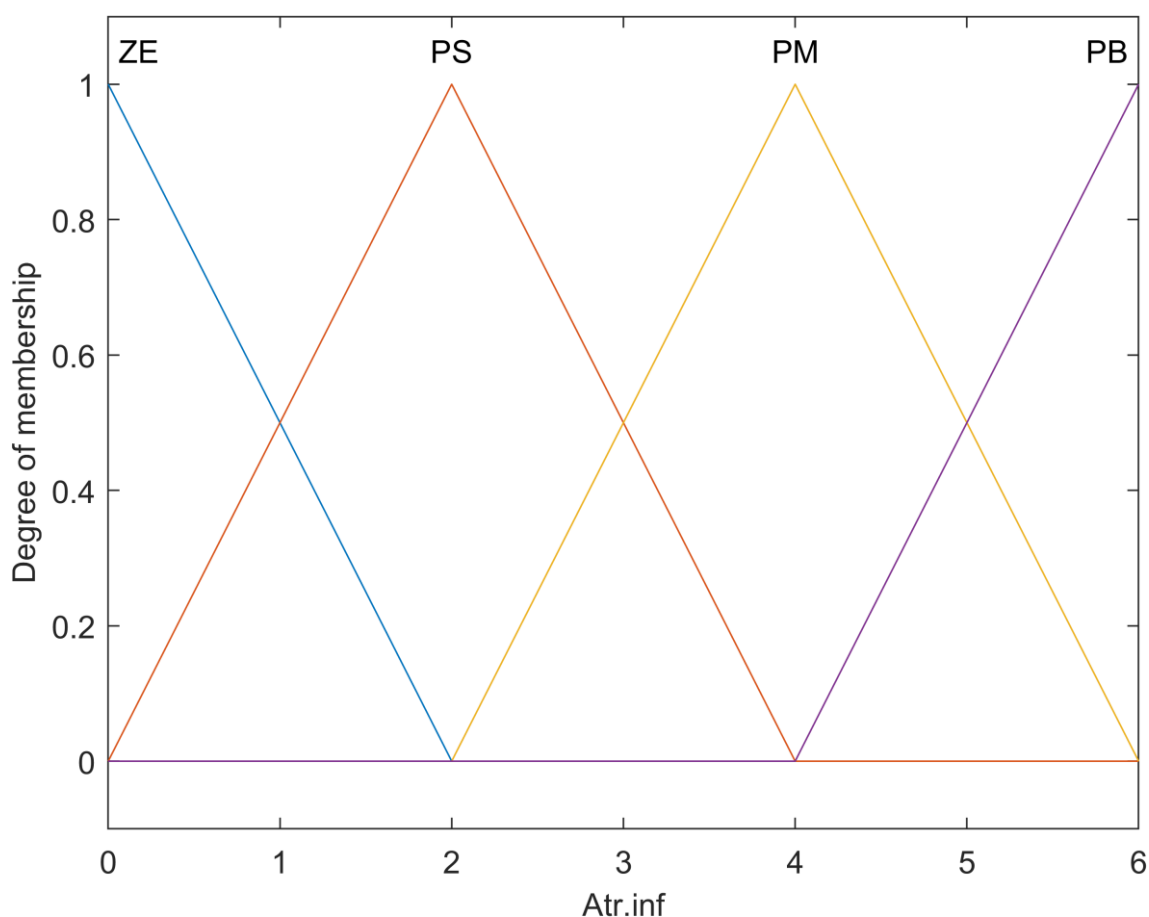
A figura acima mostra os sete valores de linguísticos seguindo a convenção de Simões e Shaw (2007) mostrada na seção 2.1.4 **Variável Linguística e Valor Linguístico**. São divididos em NB (grande negativo) até PB (grande positivo). Todos são triangulares com um universo de discurso X de $[-6, 6]$. Para essa figura, em específico, os valores linguísticos representados por funções de pertinência (*Membership Functions*) relacionam um valor *crisp* da entrada Erro do Relaxamento Muscular (RM.e) com seu grau de pertinência (*Degree of Membership*), mas esse processo é repetido para cada entrada.

Cada função de pertinência pode ser gerada a partir da Equação 11 na seção 2.1.5 **Função de Pertinência** utilizando os parâmetros a , b e c . Esses são, respectivamente, quando o triângulo começa, o pico e o final. No MATLAB foram utilizadas funções com os parâmetros mostrado na Tabela 2 seguindo o que foi feito em Shieh *et al.* (2009).

Tabela 2 – Parâmetros das funções de pertinências para a fuzzificação

Valor linguístico	<i>a</i>	<i>b</i>	<i>c</i>	Parâmetros MATLAB
NB	-6	-6	-4	[-6,-6,-4]
NM	-6	-4	-2	[-6,-4,-2]
NS	-4	-2	0	[-4,-2,0]
ZE	-2	0	2	[-2,0,2]
PS	0	2	4	[0,2,4]
PM	2	4	6	[2,4,6]
PB	4	6	6	[4,6,6]

Fonte: Shieh *et al.* (2009)

Figura 15 – Valores linguísticos das saídas dos controladores T1.FLC

Fonte: Adaptada de Shieh *et al.* (2009)

A Figura 15 mostra os quatro valores de linguísticos de maneira similar aos valores das entradas, porém, não existe saídas negativas, porque são doses de remédios injetáveis e inaláveis, assim não podendo fazer o caminho inverso. Todos são triangulares com um universo de discurso X de $[0,6]$. Para essa figura, especificamente, a Infusão de

Atracúrio (Atr.inf) está sendo relacionada a um grau de pertinência, mas o processo mesmo para cada saída.

De maneira similar às funções de pertinência dos valores de entrada, os parâmetros utilizados no MATLAB estão na Tabela 3.

Tabela 3 – Parâmetros das funções de pertinências para a defuzzificação

Valor linguístico	<i>a</i>	<i>b</i>	<i>c</i>	Parâmetros MATLAB
ZE	0	0	2	[0,0,2]
PS	0	2	4	[0,2,4]
PM	2	4	6	[2,4,6]
PB	4	6	6	[4,6,6]

Fonte: Shieh *et al.* (2009)

A base de conhecimento para o controlador com duas entradas será composta por duas bases de regras *fuzzy* mostradas nos Quadro 3 e Quadro 4 seguindo o descrito em Shieh *et al.* (2009) e a estrutura na seção 2.2.2 **Base de Conhecimento**. Além disso, o Quadro 2 demonstra como ler as bases de regras para duas entradas, explicando a estrutura básica para uma combinação de valores linguísticos de Grande Negativo (NB) até Grande Positivo (PB) da Entrada 1 e Entrada 2 para um valor linguístico de Zero (ZE) até PB para a Saída.

Quadro 2 – Estrutura das bases de regras dos controladores com duas entradas

Saída	Valor Linguístico da Entrada 1 (NB, NM, NS, ZE, PS, PM ou PB)
Valor Linguístico da Entrada 2 (NB, NM, NS, ZE, PS, PM ou PB)	Valor Linguístico da Saída (ZE, PS, PM, PB)

Quadro 3 – Base de regras para a saída Atr.inf no controlador com duas entradas

Atr.inf	NB	NM	NS	ZE	PS	PM	PB
NB	PB		PS		ZE		ZE
NM		PM		ZE		ZE	
NS	PB		PS		ZE		ZE
ZE		PM		ZE		ZE	
PS	PM		ZE		ZE		ZE
PM		PM		ZE		ZE	
PB	PM		ZE		ZE		ZE

Fonte: Adaptado de Shieh *et al.* (2009)

Quadro 4 – Base de regras para a saída Iso.conc no controlador com duas entradas

Iso.conc	NB	NM	NS	ZE	PS	PM	PB
NB	ZE		ZE		ZE		ZE
NM		ZE		ZE		ZE	
NS	ZE		ZE		ZE		ZE
ZE		ZE		ZE		ZE	
PS	PS		PS		PS		ZE
PM		PM		PS		ZE	
PB	PB		PM		PS		PS

Fonte: Adaptado de Shieh *et al.* (2009)

O Quadro 3 mostra de maneira simplificada cada regra do controlador com duas entradas para a saída Infusão de Atracúrio (Atr.inf). Para sete valores linguísticos de NB até PB e duas entradas, é possível ter até 49 regras distintas, mas são descritas apenas 25 regras nas células em cinza-claro com um valor linguístico da saída de ZE até PB. São representados os valores das entradas 1 e 2 na primeira, respectivamente, linha e coluna em negrito para um valor da saída na célula em cinza-claro, logo, por exemplo, uma das regras seria: se Entrada 1 for **NM** e Entrada 2 for **ZE** então Saída Atr.inf é **PM**.

Especificando mais a regra anterior, nesse caso, se o Erro do Relaxamento Muscular (**RM.e**) tiver em média maior grau de pertinência no valor linguístico **NM** e o Erro da Pressão Sanguínea (**PS.e**) for mais pertinente ao valor **ZE**, então a saída de Infusão de Atracúrio (**Atr.inf**) terá uma maior pertinência, em média, no valor **PM**. Numericamente é preciso utilizar os valores de escalares na Tabela 4 que serão melhor explicados no final desta seção, então, um valor de **RM.e** de aproximadamente -0,13 terá um grau 1 de pertinência no valor **NM**, enquanto para o **PS.e** com valor 0 terá o mesmo para **ZE**. Isso levaria a saída a ter, aproximadamente 2. Qualquer regra da base de regras dos controladores *fuzzy* terá esse comportamento, seguindo todos os passos de fuzzificação e inferência.

De maneira similar ao Quadro 3, o Quadro 4 mostra de maneira simplificada cada regra do controlador com duas entradas para a saída Concentração do Isoflurano (Iso.conc). Um exemplo de uma das regras seria: se Entrada 1 for **NS** e Entrada 2 for **PB** então Saída Iso.conc é **PM**.

O controlador com quatro entradas terá maior possibilidades de regras diferentes. Agora, a base regras terá seis vezes mais regras, porque esse é o número de combinações possíveis de quatro entradas em duas não importando a ordem, ou seja, a tabela dirá qual é a saída para a entrada 1 e 2 até para a entrada 3 e 4. Isso acontece porque o controlador

é decomposto em vários sub-controladores, explicado mais adiante na seção 3.3.3 **Controlador com quatro entradas e duas saídas (T1.42.FLC)**.

As bases de regras para o controlador estão mostradas no Quadro 6 e Quadro 7 seguindo Shieh *et al.* (2009) e a estrutura da regra na seção 2.2.2 **Base de Conhecimento**, enquanto a Quadro 5 descreve como ler as bases de regras com seis combinações diferentes.

Quadro 5 – Estrutura das bases de regras dos controladores com quatro entradas

Saída	Valor Linguístico da Entrada A (NB, NM, NS, ZE, PS, PM ou PB)	
Valor Linguístico da Entrada B (NB, NM, NS, ZE, PS, PM ou PB)	Combinação 12 (AB)	Combinação 13 (AB)
	Combinação 14 (AB)	Combinação 23 (AB)
	Combinação 24 (AB)	Combinação 34 (AB)

Fonte: Elaborado pelo Autor (2022)

Quadro 6 – Base de regras para a saída Atr.inf no controlador com quatro entradas

Atr.inf	NB		NM		NS		ZE		PS		PM		PB	
NB	ZE	PB			ZE	PM			ZE	ZE			ZE	ZE
	PB	PB			PS	PS			ZE	ZE			ZE	ZE
	PB	PB			PM	PS			ZE	ZE			ZE	ZE
NM			ZE	PM			ZE	ZE			ZE	ZE		
			PM	PM			ZE	ZE			ZE	ZE		
			PM	PM			ZE	ZE			ZE	ZE		
NS	ZE	PM			ZE	PM			ZE	ZE			ZE	ZE
	PB	PB			PM	PM			ZE	ZE			ZE	ZE
	PB	PB			PM	PM			ZE	ZE			ZE	ZE
ZE			ZE	PM			ZE	ZE			ZE	ZE		
			PM	PM			ZE	ZE			ZE	ZE		
			PS	PM			ZE	ZE			ZE	ZE		
PS	ZE	PS			ZE	PS			ZE	ZE			ZE	ZE
	PM	PM			ZE	ZE			ZE	ZE			ZE	ZE
	PB	PB			PS	PS			ZE	ZE			ZE	ZE
PM			ZE	PS			ZE	ZE			ZE	ZE		
			PM	PM			ZE	ZE			ZE	ZE		
			PS	PM			ZE	ZE			ZE	ZE		
PB	ZE	PS			ZE	PS			ZE	ZE			ZE	ZE
	PM	PB			ZE	ZE			ZE	ZE			ZE	ZE
	PB	PB			PS	PS			ZE	ZE			ZE	ZE

Fonte: Adaptado de Shieh *et al.* (2009)

O Quadro 5 apresenta como deve ser lido o Quadro 6 e o Quadro 7, ou seja, as bases de regras do controlador com quatro entradas. Cada número representa uma

combinação de duas entradas A e B (AB), como entrada 1 com 2 (12) ou entrada 3 com 4 (34) e essas serão substituídas por valores linguísticos da saída correspondente na base.

O Quadro 6 descreve a maneira simplificada cada regra do controlador com quatro entradas para a saída Infusão de Atracúrio (Atr.inf). São 125 regras descritas para seis combinações de entradas representando um sub-controlador. A leitura é similar à das bases de regras dos controladores de duas entradas no Quadro 3 e Quadro 4, juntando com o guia mostrado no Quadro 5, assim, por exemplo, uma das regras está descrita como: se Entrada 1 for **NS** e Entrada 3 for **PS** então Saída Atr.inf é **PS**.

Para finalizar as descrições das bases de regras, o Quadro 7 mostra cada regra do controlador com quatro entradas para a saída Concentração de Isoflurano (Iso.conc). De maneira similar ao Quadro 6, um exemplos de regra descrita é: se Entrada 3 for **ZE** e Entrada 4 for **PM** então Saída Iso.conc é **PS**.

Quadro 7 – Base de regras para a saída Iso.conc no controlador com quatro entradas

Iso.conc	NB		NM		NS		ZE		PS		PM		PB	
NB	ZE	PB			ZE	PM			ZE	ZE			PS	ZE
	ZE	ZE			ZE	ZE			ZE	ZE			ZE	ZE
	ZE	ZE			ZE	ZE			ZE	ZE			ZE	ZE
NM			ZE	PM			ZE	ZE			PS	ZE		
			ZE	ZE			ZE	ZE			ZE	ZE		
			ZE	ZE			ZE	ZE			ZE	ZE		
NS	ZE	PS			ZE	PS			PS	ZE			PM	ZE
	ZE	ZE			ZE	ZE			ZE	ZE			ZE	ZE
	ZE	ZE			ZE	ZE			ZE	ZE			ZE	ZE
ZE			ZE	ZE			ZE	ZE			PM	ZE		
			ZE	ZE			ZE	ZE			ZE	ZE		
			ZE	ZE			ZE	ZE			ZE	ZE		
PS	ZE	ZE			PS	ZE			PS	ZE			PB	ZE
	PS	PS			PS	PS			PS	PS			ZE	ZE
	PS	PM			PS	PS			PS	ZE			ZE	ZE
PM			PS	ZE			PM	ZE			PB	ZE		
			PM	PM			PS	PS			ZE	ZE		
			PM	PM			PS	PS			ZE	PS		
PB	PS	ZE			PM	ZE			PB	ZE			PB	ZE
	PB	PB			PM	PM			PS	PS			PS	PS
	PB	PB			PM	PM			PS	PS			PS	PS

Fonte: Adaptado de Shieh *et al.* (2009)

As especificações da inferência escolhida são dadas pelo modelo Mamdani na Tabela 1 e serão utilizadas de acordo com o especificado na seção 2.2.3 **Inferência**. Os

métodos para calcular a união e interseção estão descritos, respectivamente, nas Equação 7 e Equação 9 na seção 2.1.3.1 União e 2.1.3.2 Interseção.

As entradas e saídas dos controladores são *crisp*, ou seja, não são valores *fuzzy*, por isso é necessário utilizar valores escalares para normalizar no universo de discurso. Por exemplo, uma entrada qualquer poderá ser de -100 a 100, mas o universo de discurso padrão dos valores linguísticos são de -6 até 6, então é utilizado um valor escalar de 0,06 para multiplicar antes de ser processado na função de pertinência e acontece o inverso nas saídas *fuzzy* para *crisp*. No projeto, serão utilizados escalares para cada entrada e saída dos dois controladores, um com duas entradas e outro com quatro, que estão descritos na Tabela 4. Esses foram escolhidos por meio de tentativa e erro, dando a melhor resposta possível para cada controlador. Além disso, os valores resultantes das entradas serão saturados para que não tenha valores acima ou abaixo do universo de discurso.

Tabela 4 – Escalares utilizados nos controladores T1.FLC

Escalar	Controlador com duas entradas	Controlador com quatro entradas
Entrada 1	30	60
Entrada 2	0,6	0,8
Entrada 3	-	0,6
Entrada 4	-	0,8
Saída 1	0,5	0,5
Saída 2	0,7	0,13

Fonte: Elaborada pelo Autor (2022)

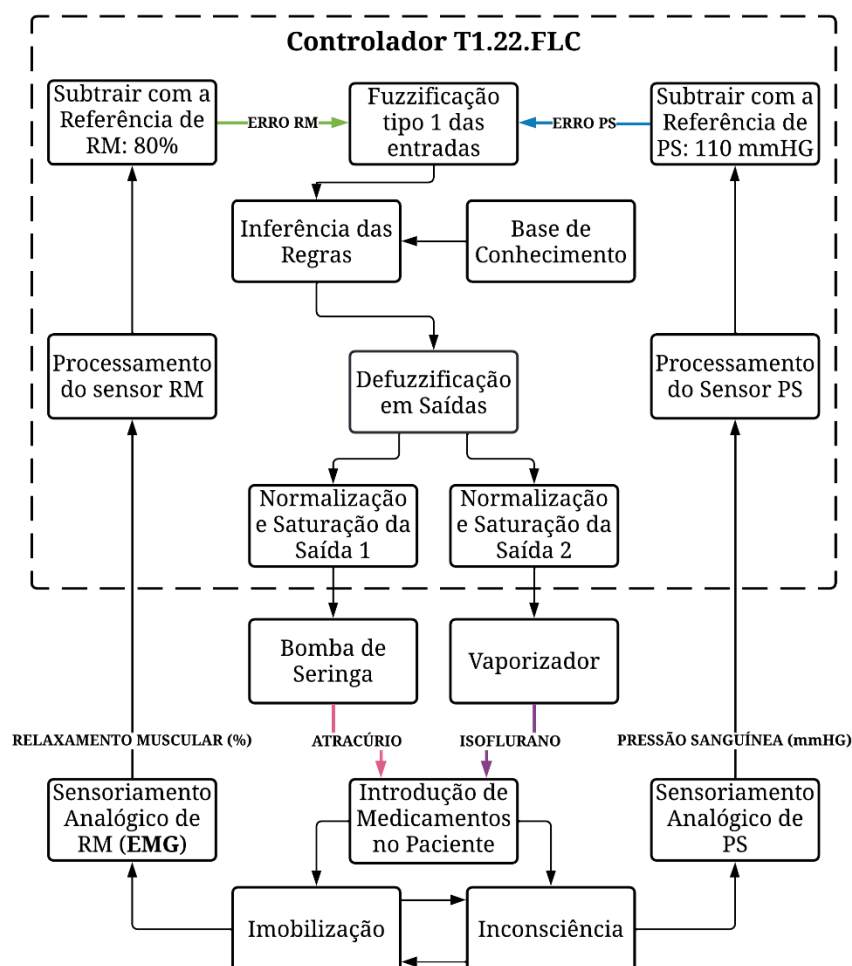
Por último, o método de defuzzificação escolhido foi o centro de área, descrito pela Equação 17 na seção 2.2.4 **Defuzzificação**.

3.3.2 Controlador com duas entradas e duas saídas (T1.22.FLC)

Esse controlador segue a estrutura de um controle *fuzzy* multivariável em malha fechada com duas entradas: Erro do Relaxamento Muscular (RM.e) e Erro da Pressão Sanguínea (PS.e) e duas saídas: Infusão de Atracúrio (Atrac.inf) e Concentração de Isoflurano (Iso.conc) mostrado em Shieh *et al.* (2009). Foram projetados dois sub-controladores, um para cada saída com base de regras separadas mas idênticas, utilizando o método de decomposição citado na subseção anterior, também seguindo o autor citado anteriormente.

A Figura 16 mostra o diagrama funcional do controle *fuzzy* tipo-1 com duas entradas e duas saídas. A Figura 17 representa o diagrama da decomposição do controlador em dois sub-controladores de duas entradas e uma saída.

Figura 16 – Diagrama funcional do controlador T1.22.FLC

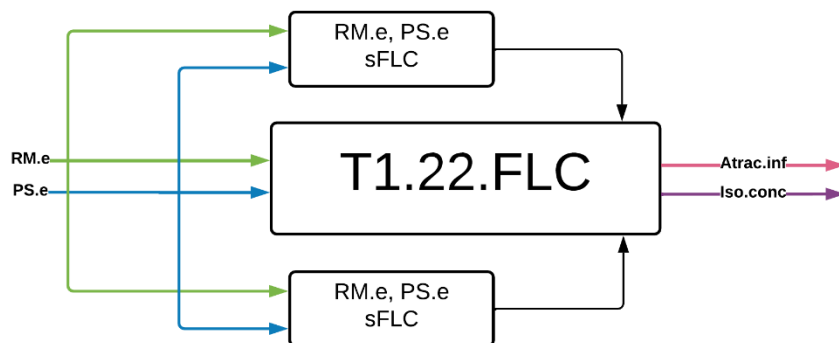


Fonte: Elaborada pelo Autor (2021)

Na Figura 16, o diagrama demonstra o processo de controle funcional da anestesia utilizando o controlador T1.22.FLC com duas entradas: Erro RM (RM.e) em verde e Erro PS (PS.e) em azul e duas saídas: Atracúrio (Atrac.inf) em rosa e Isoflurano (Iso.conc) em roxo. O bloco com linhas pontilhadas determina o controlador *fuzzy* com funções típicas já mencionadas anteriormente: Fuzzificação, Inferência de Regras + Base de Conhecimento e Defuzzificação. Os blocos de Subtração da entrada, Processamento dos sensores e Normalização da saída são funções de processamento do próprio sistema computacional em que abriga o algoritmo de computador do controlador. Já os blocos dos

atuadores: Bomba de Seringa e Vaporizador estão fora do controlador e computador, mas produzem a saída física necessária para o controle. O bloco de Introdução dos medicamentos faz com que os objetivos da anestesia para esse controle, Imobilização e Inconsciência, sejam atingidos. Por último os blocos de Sensoriamento retornam o valor medido de volta ao controlador para o controle em malha fechada.

Figura 17 – Diagrama de decomposição do controlador T1.22.FLC



Fonte: Adaptada de Shieh *et al.* (2009)

A Figura 17 mostra o diagrama de decomposição multivariável do controlador T2.22.FLC, que é funcionalmente dividido em dois sub-controladores (sFLC) com bases de regras diferentes de acordo com Shieh *et al.* (2009). A saída de cada sub-controlador será uma saída do controlador principal: Atrac.inf em rosa ou Iso.conc. em roxo, assim reduzindo de duas entradas e duas saídas para dois com duas entradas e uma saída.

3.3.3 Controlador com quatro entradas e duas saídas (T1.42.FLC)

Neste controlador são adicionadas duas entradas a mais em relação ao T1.22.FLC: Integral do Erro RM (i.RM.e) e Integral do Erro PS (i.PS.e). Assim, o controlador possui quatro entradas e duas saídas. Para decompor em vários sub-controladores de duas entradas e uma saída é utilizada uma combinação simples de quatro entradas para duas, sem importar a ordem:

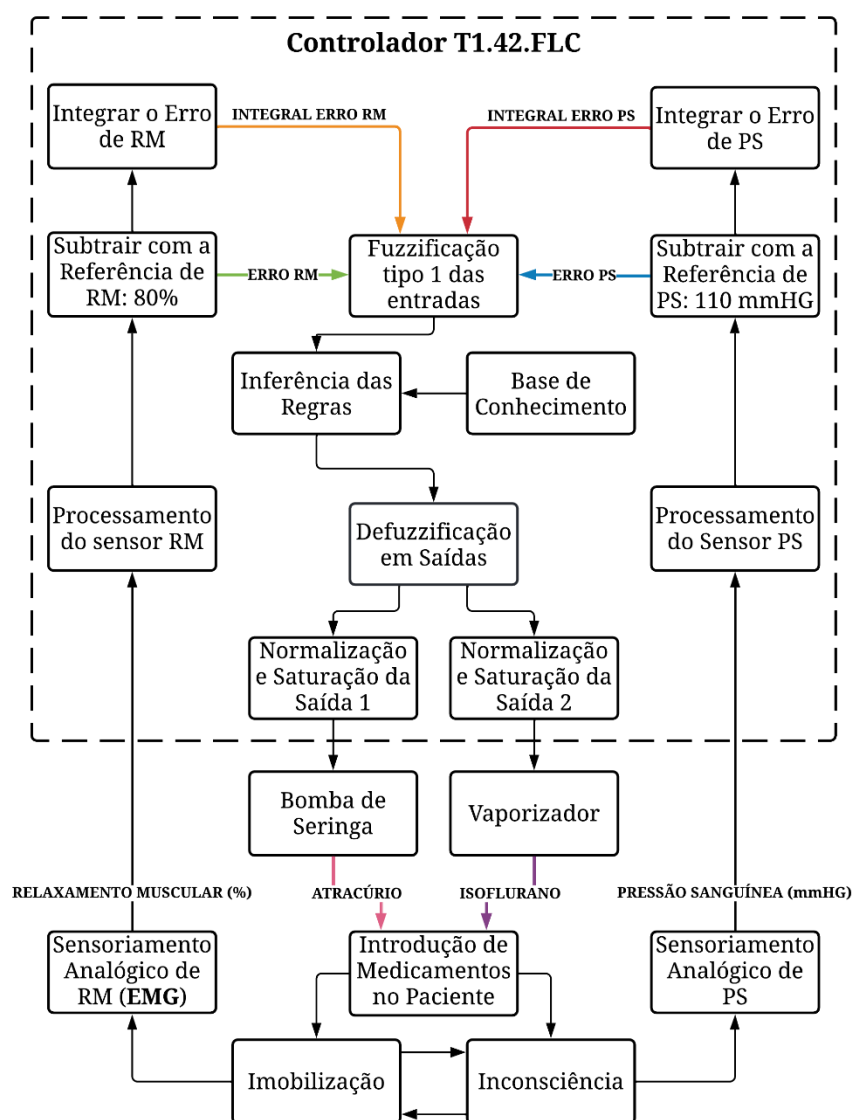
$$C_k^n = \frac{n!}{k! (n - k)!} \quad (28)$$

$$C_2^4 = \frac{4!}{2! (4 - 2)!} = \frac{4!}{2! 2!} = 6 \quad (29)$$

São seis sub-controladores para cada saída, totalizando em doze, cada um com base de regras diferentes extraídas de Shieh *et al.* (2009).

A Figura 18 mostra o diagrama funcional do controle *fuzzy* tipo-1 com quatro entradas e duas saídas. A Figura 19 representa o diagrama da decomposição do controlador em doze sub-controladores de duas entradas e uma saída.

Figura 18 – Diagrama funcional do controlador T1.42.FLC

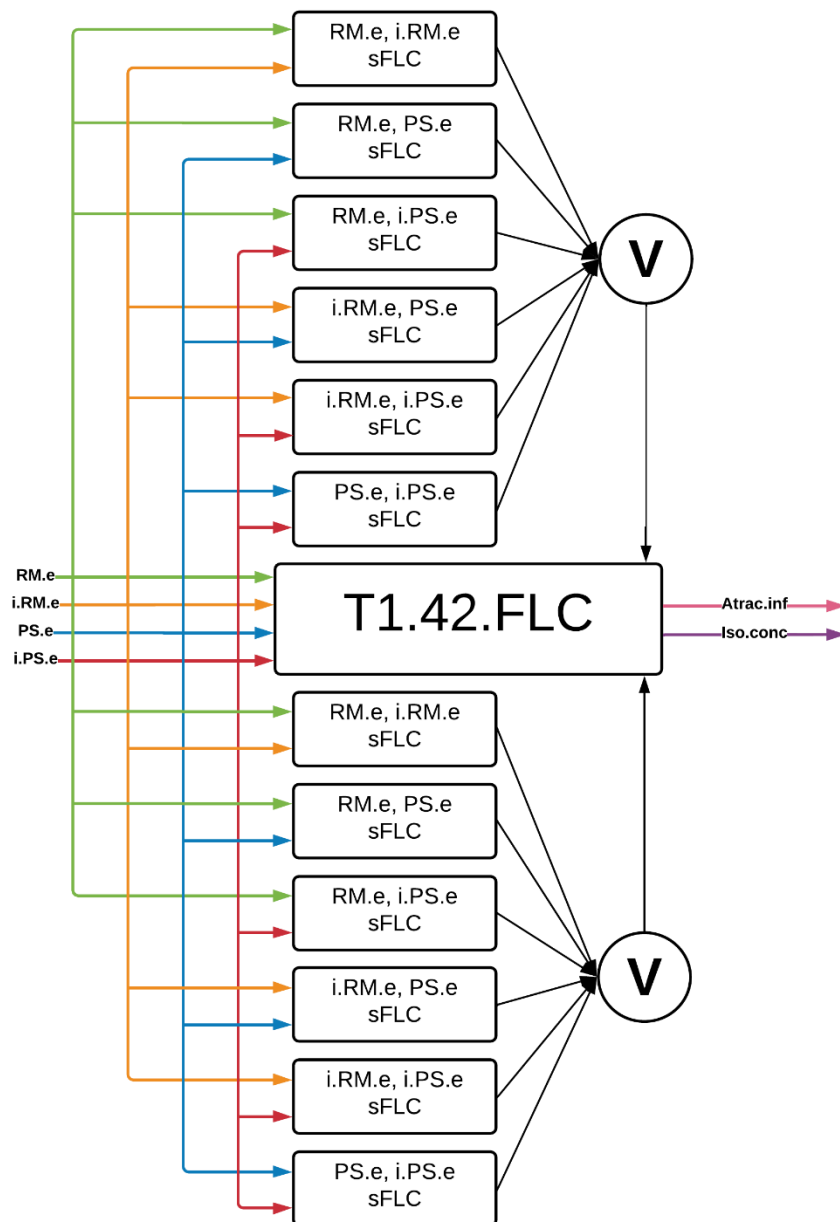


Fonte: Elaborada pelo Autor (2021)

A Figura 18 demonstra o diagrama funcional do processo de controle da anestesia utilizando o controlador T1.42.FLC. As duas novas entradas são: Integral Erro RM (i.RM.e) em laranja e Integral Erro PS (i.PS.e) em vermelho. O bloco com linhas

pontilhadas determina o controlador *fuzzy* com blocos similares ao T1.22.FLC, com a adição de blocos de integração numérica realizada pelo sistema computacional.

Figura 19 – Diagrama de decomposição do controlador T1.42.FLC



Fonte: Adaptada de Shieh *et al.* (2009)

Na Figura 19, mostra-se a decomposição do T1.42.FLC em doze sub-controladores (sFLC), com seis para cada saída. O operador V retorna o valor máximo de seis sFLC para cada saída final.

3.4 PROJETO DE CONTROLADORES FUZZY TIPO-2 INTERVALAR (IT2.FLC)

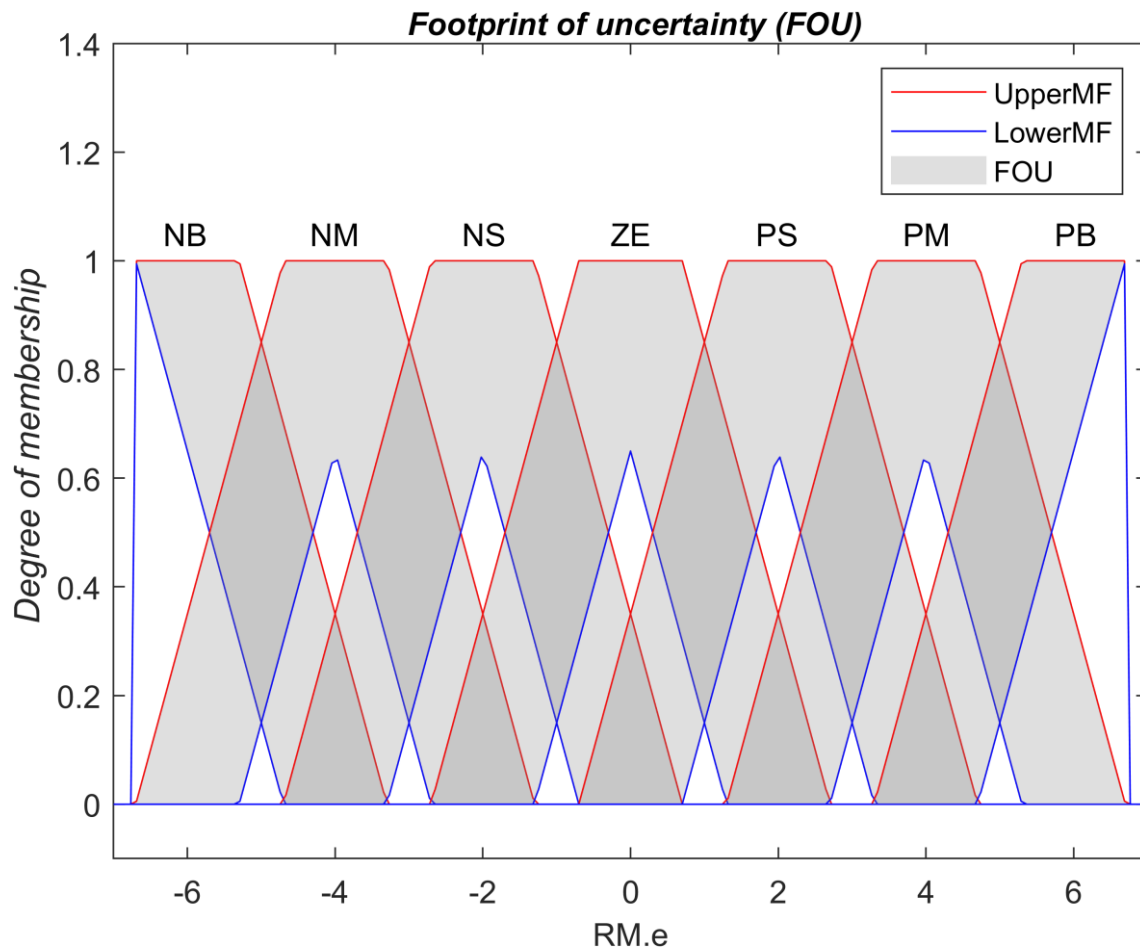
Esses controladores utilizarão a lógica *fuzzy* tipo-2 intervalar para controlar a anestesia geral. As variáveis controladas, Relaxamento Muscular (RM) e Pressão Sanguínea (PS), são utilizadas para encontrar o erro comparando com a referência. O controle *fuzzy* é feito logo em seguida, seguindo os passos de: fuzzificação; inferência de regras a partir da base de conhecimento; tipo-redutor e defuzzificação. As saídas do controlador serão multiplicadas para alguma escala necessária para os atuadores, sendo que esses irão atuar no paciente, assim modificando seus sinais vitais e levando até a imobilidade, pelo aumento do relaxamento muscular, e inconsciência, pela diminuição da pressão sanguínea.

Nas seções seguintes, será descrito as especificações de projeto dos controladores *fuzzy* tipo-2 intervalar e dois tipos de controladores projetados, um deles com duas entradas e outro com quatro entradas, incluindo diagramas funcional e de decomposição.

3.4.1 Especificações de projeto dos controladores IT2.FLC

O projeto de controladores *fuzzy* tipo-2 intervalar seguirá a estrutura mostrada na Figura 6. Cada bloco representa o funcionamento do IT2.FLC.

A fuzzificação e defuzzificação utilizarão funções de pertinências triangulares para representação dos valores linguísticos de maneira semelhante aos controladores T1.FLC (Figura 14 e Figura 15), porém, agora será definida a mancha de incerteza ou *Footprint Of Uncertainty* (FOU) intervalar para cada entrada e cada saída, ou seja, será adicionado e subtraído um valor da função, assim criando uma incerteza na definição do valor linguístico. O universo de discurso será $[-7,7]$ para as entradas e $[0,7]$ para as saídas para acomodar a incerteza. Foi utilizado o método descrito em Doctor *et al.* (2016) partindo de dados clínicos de sinais dos pacientes em anestesia para obter a FOU das duas entradas principais: RM.e e PS.e. Os valores linguísticos da primeira entrada encontram-se na Figura 20.

Figura 20 – Valores linguísticos da entrada RM.e dos controladores IT2.FLC

Fonte: Adaptada de Doctor *et al.* (2016)

A Figura 20 mostra os valores linguísticos da entrada RM.e com funções pertinência triangulares com um FOU calculado. No MATLAB é definida a função de pertinência (MF) inferior em azul e a MF superior em vermelho com o FOU em cinza. O objetivo é utilizar somente MF triangular somando e subtraindo o FOU para formar o conjunto *fuzzy* tipo-2 intervalar, mas para funcionar no *software* é necessário definir a função de pertinência superior como trapezoidal e utilizar a FOU para criar a MF inferior, criando o efeito desejado.

As funções de pertinência referentes à fuzzificação das duas entradas não são idênticas, mas partem do mesmo princípio, utilizando valores linguísticos como base (Figura 14) e utilizar um FOU calculado para formar os conjuntos *fuzzy* tipo-2 intervalares. A Tabela 5 mostra os valores e parâmetros utilizados, junto com os mesmos parâmetros a , b e c , ou seja, respectivamente o parâmetro de começo, pico e fim do

triângulo discutidos na seção 3.3.1 **Especificações de projeto dos controladores T1.FLC** e com os valores mostrados na Tabela 2.

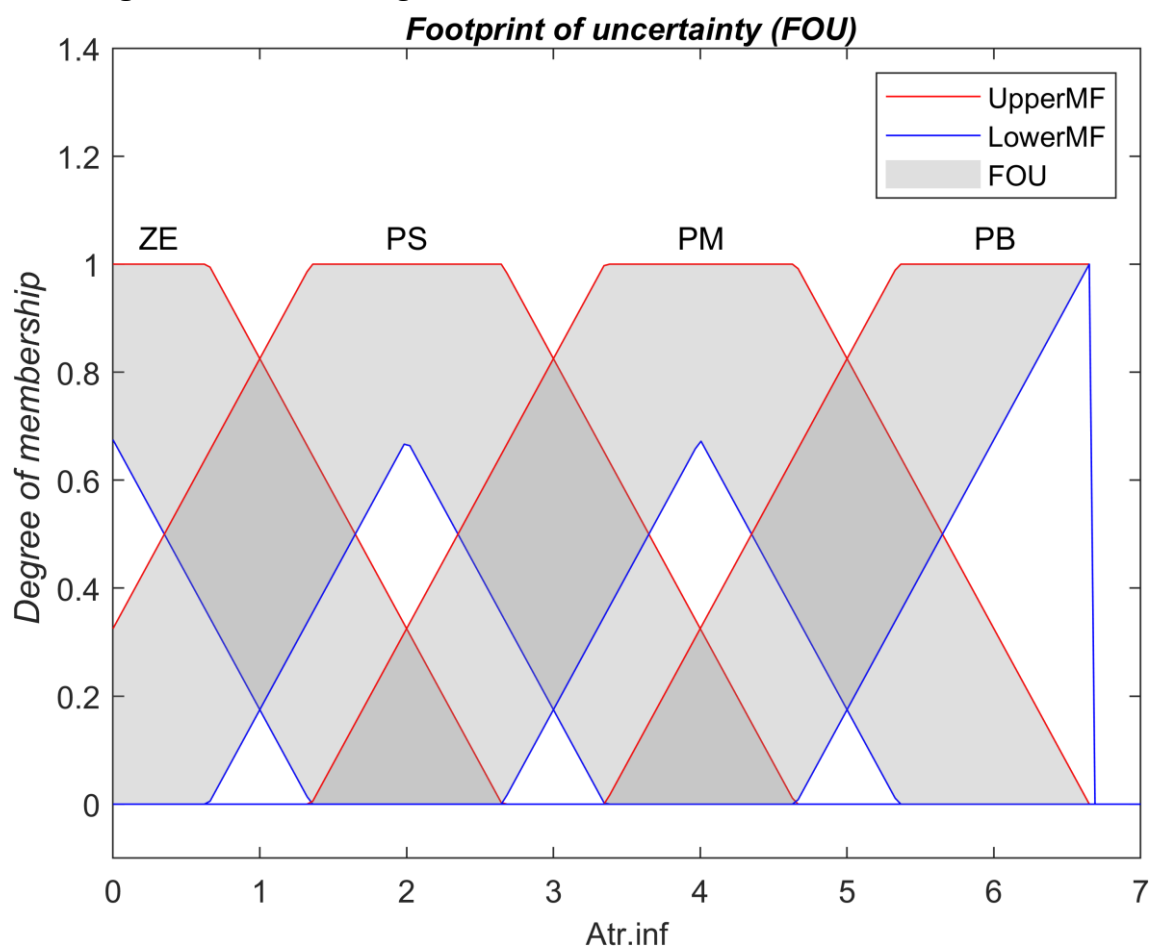
Tabela 5 – Parâmetros das funções de pertinência das entradas dos IT2.FLC

Entradas	FOU	Parâmetros MF superior MATLAB
RM.e	$\pm 0,7$	$[(a - 0,7), (b - 0,7), (b + 0,7), (c + 0,7)]$
PS.e	± 1	$[(a - 1), (b - 1), (b + 1), (c + 1)]$

Fonte: Doctor *et al.* (2016)

Para a defuzzificação das saída: Atr.inf e Iso.conc, de maneira similar à fuzzificação, será utilizado um FOU para cada, porém, esse é definido por tentativa e erro. A Figura 21 mostra os valores linguísticos da primeira saída.

Figura 21 – Valores linguísticos da saída Atr.inf dos controladores IT2.FLC



Fonte: Adaptada de Doctor *et al.* (2016)

De maneira similar à Figura 20 e Figura 15, a figura anterior mostra os valores linguísticos da saída Atr.inf com MF superior, inferior e FOU. Também é mostrada na

Tabela 6, de maneira similar à Tabela 5, os parâmetros utilizados para gerar os conjuntos *fuzzy* tipo-2 intervalares das saídas.

Tabela 6 – Parâmetros das funções de pertinência das saídas dos IT2.FLC

Saídas	FOU	Parâmetros MF superior MATLAB
Atr.inf	$\pm 0,65$	$[(a - 0,65), (b - 0,65), (b + 0,65), (c + 0,65)]$
Iso.conc	$\pm 0,1$	$[(a - 0,1), (b - 0,1), (b + 0,1), (c + 0,1)]$

Fonte: Elaborada pelo Autor (2022)

As mesmas bases de regras descritas no Quadro 3, Quadro 4, Quadro 6 e Quadro 7 são utilizadas para os controladores IT2.FLC com duas e quatro entradas.

Para a inferência e defuzzificação, também serão utilizados os mesmos métodos descritos na seção 3.3.1 **Especificações de projeto dos controladores T1.FLC**, enquanto o método de tipo-redutor utilizado é o EIASC, que foi citado na seção 2.2.5.1 Tipo-redutor.

A Tabela 7, com um conteúdo similar ao da Tabela 4, mostra os escalares utilizados nos controladores IT2.FLC.

Tabela 7 – Escalares utilizados nos controladores IT2.FLC

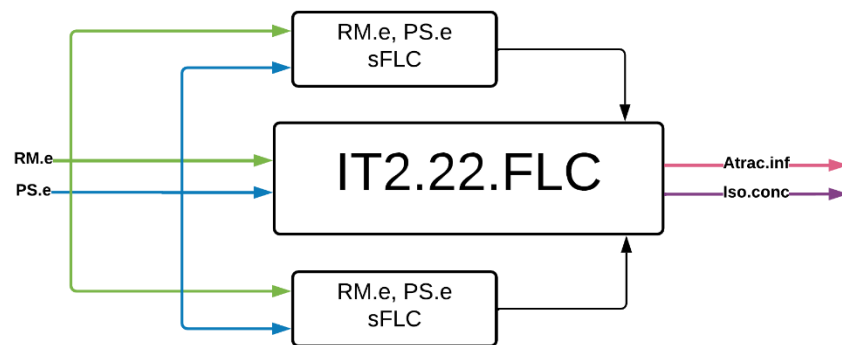
Escalar	Controlador com duas entradas	Controlador com quatro entradas
Entrada 1	140	17,5
Entrada 2	0,875	0,1
Entrada 3	-	4,67
Entrada 4	-	0,85
Saída 1	0,55	0,65
Saída 2	0,7	0,15

Fonte: Elaborada pelo Autor (2022)

3.4.2 Controlador com duas entradas e duas saídas (IT2.22.FLC)

De maneira similar ao T1.22.FLC, esse controlador seguirá todos os passos de um tipo-1 com duas entradas (RM.e; PS.e), porém, com um passo adicional chamado de tipo-redutor e utilizando conjuntos *fuzzy* tipo-2 intervalares. O seu diagrama funcional é demonstrado na Figura 22.

Figura 23 – Diagrama de decomposição do controlador IT2.22.FLC

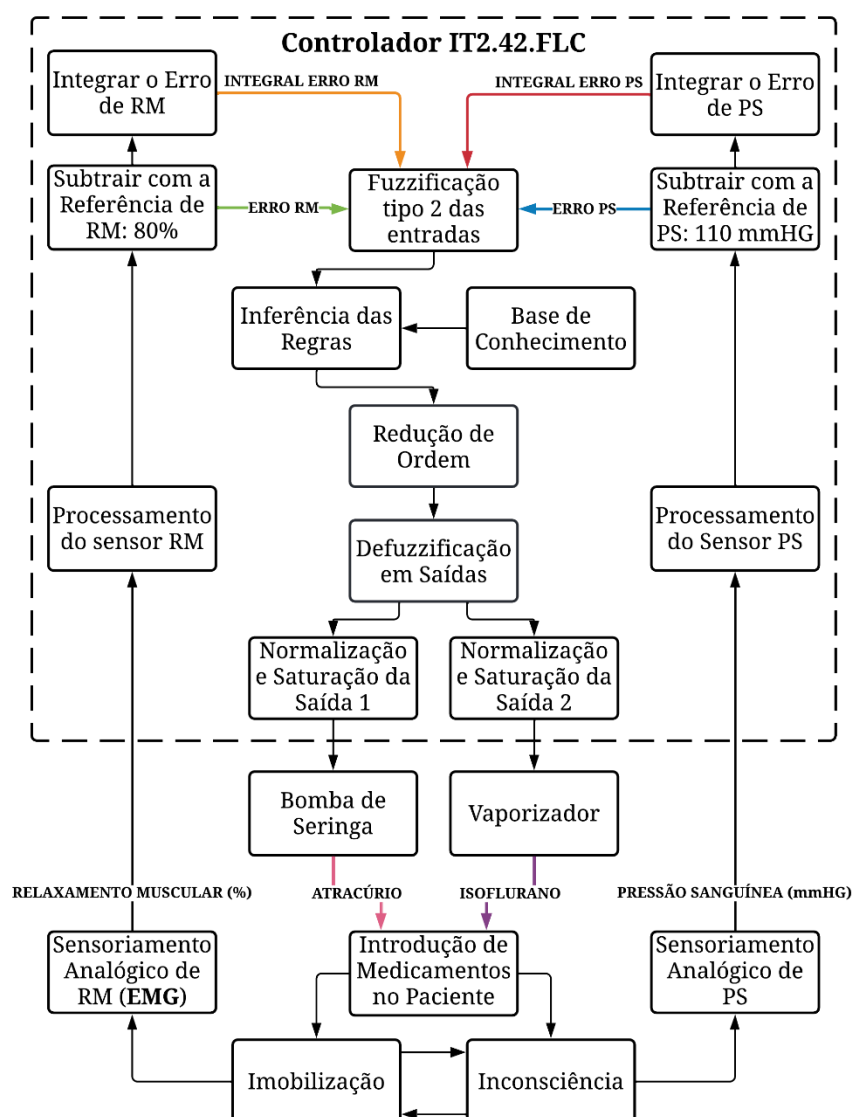


Fonte: Adaptada de Doctor *et al.* (2016)

Na Figura 23, o diagrama de decomposição contém dois sFLC com lógica *fuzzy* tipo-2 que calculam cada saída do IT2.22.FLC.

3.4.3 Controlador com quatro entradas e duas saídas (IT2.42.FLC)

Este controlador inclui mais duas saídas (i.RM.e; i.PS.e) por meio de um bloco de integração das saídas do IT2.22.FLC. Seu diagrama funcional está na Figura 24.

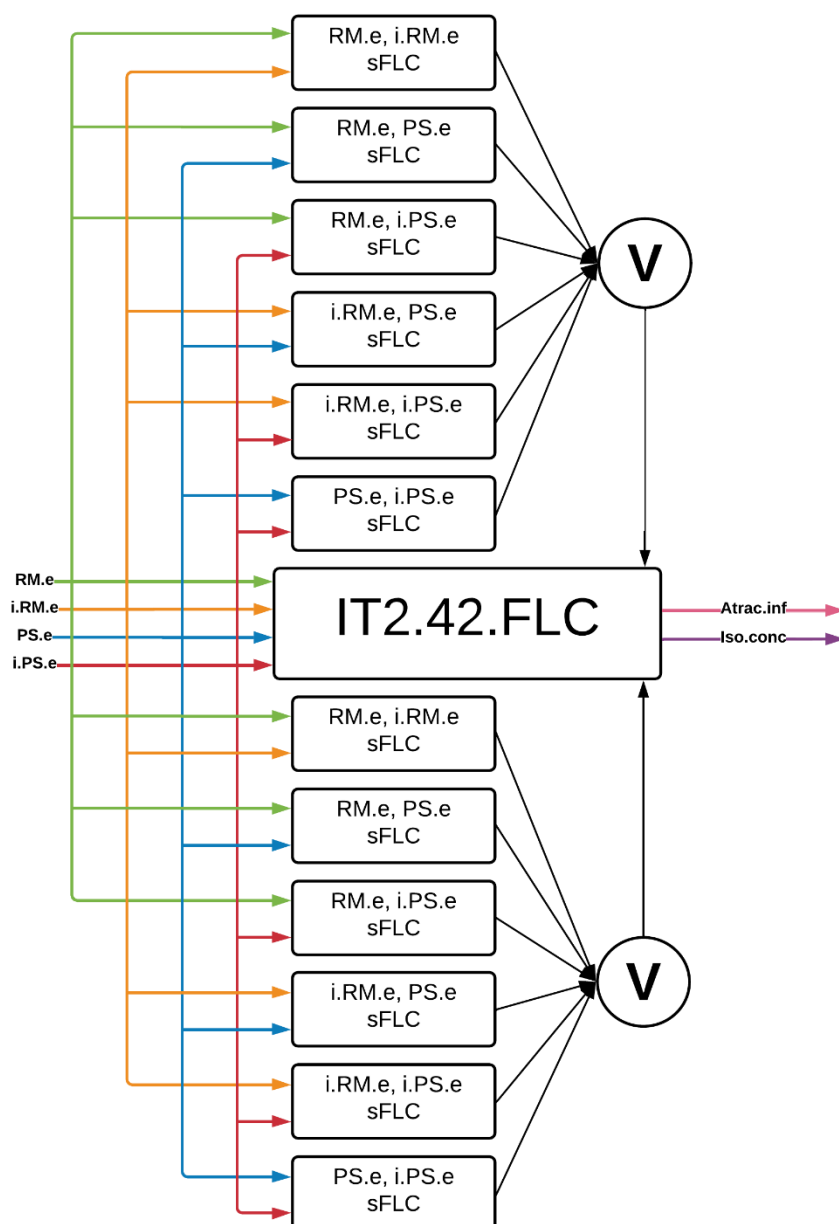
Figura 24 – Diagrama funcional do controlador IT2.42.FLC

Fonte: Elaborada pelo Autor (2021)

A Figura 24 mostra o diagrama funcional do IT2.42.FLC, ou seja, contém quatro entradas e duas saídas e utiliza lógica *fuzzy* tipo-2.

Cada combinação de duas entradas vai para um dos doze sFLC (Figura 25) de maneira similar ao T1.42.FLC.

Figura 25 – Diagrama de decomposição do controlador IT2.42.FLC



Fonte: Adaptada de Doctor *et al.* (2016)

Na Figura 25, o diagrama de decomposição mostra doze sFLC com combinações de duas entradas diferentes. O operador V representa a operação de valor máximo e está ligado a seis sFLC para cada saída do IT2.42.FLC.

4 SIMULAÇÕES E ANÁLISES DOS RESULTADOS

As simulações finais do controle de anestesia utilizando controladores com lógica *fuzzy* tipo-1 e tipo-2 irão demonstrar suas diferenças de desempenho, enquanto uma representação física do sistema servirá para visualizar a viabilidade de construção.

Neste capítulo serão demonstrados e analisados os resultados das simulações de cada controlador e da emulação do sistema físico.

4.1 SIMULAÇÕES

A simulação matemática do sistema será importante para comprovar a funcionalidade correta e verificar o desempenho dos controlador *fuzzy* tipo-1 e 2.

Será necessário para a simulação o modelo matemático de farmacocinética e farmacodinâmica que descreve a reação dos parâmetros de controle do corpo humano na anestesia (pressão sanguínea e relaxamento muscular) a partir da entrada dos fármacos (Atracúrio e Isoflurano), demonstrada na seção 2.3.4.6 Modelo Multivariável do Corpo Humano para Anestesia, já utilizado em Shieh *et al.* (2009) e Doctor *et al.* (2016).

As simulações serão feitas utilizando programas do MATLAB ou *scripts* a partir do projeto de controlador e modelo matemático junto com funções específicas para simular em tempo determinado as mudanças dinâmicas nas entradas e saídas do controlador e modelo farmacocinético e farmacodinâmico. Simulações no Simulink também serão montadas, mas somente para serem utilizadas como base para a programação dos *scripts*, porque são relativamente menos flexíveis do que códigos escritos. Os parâmetros das simulações estão descritos na tabela a seguir.

Tabela 8 – Parâmetros das simulações

Controlador	Referência RM	Referência PS	Tempo máximo
T1.22.FLC	0,8	110 mmHG	300 minutos
T1.42.FLC	0,8	110 mmHG	300 minutos
IT2.22.FLC	0,8	110 mmHG	300 minutos
IT2.42.FLC	0,8	110 mmHG	300 minutos

Fonte: Elaborada pelo Autor (2021)

Na Tabela 8, os parâmetros de simulação para cada controlador estão expostos. A referência se refere ao valor alvo para cada saída, em que o objetivo principal do

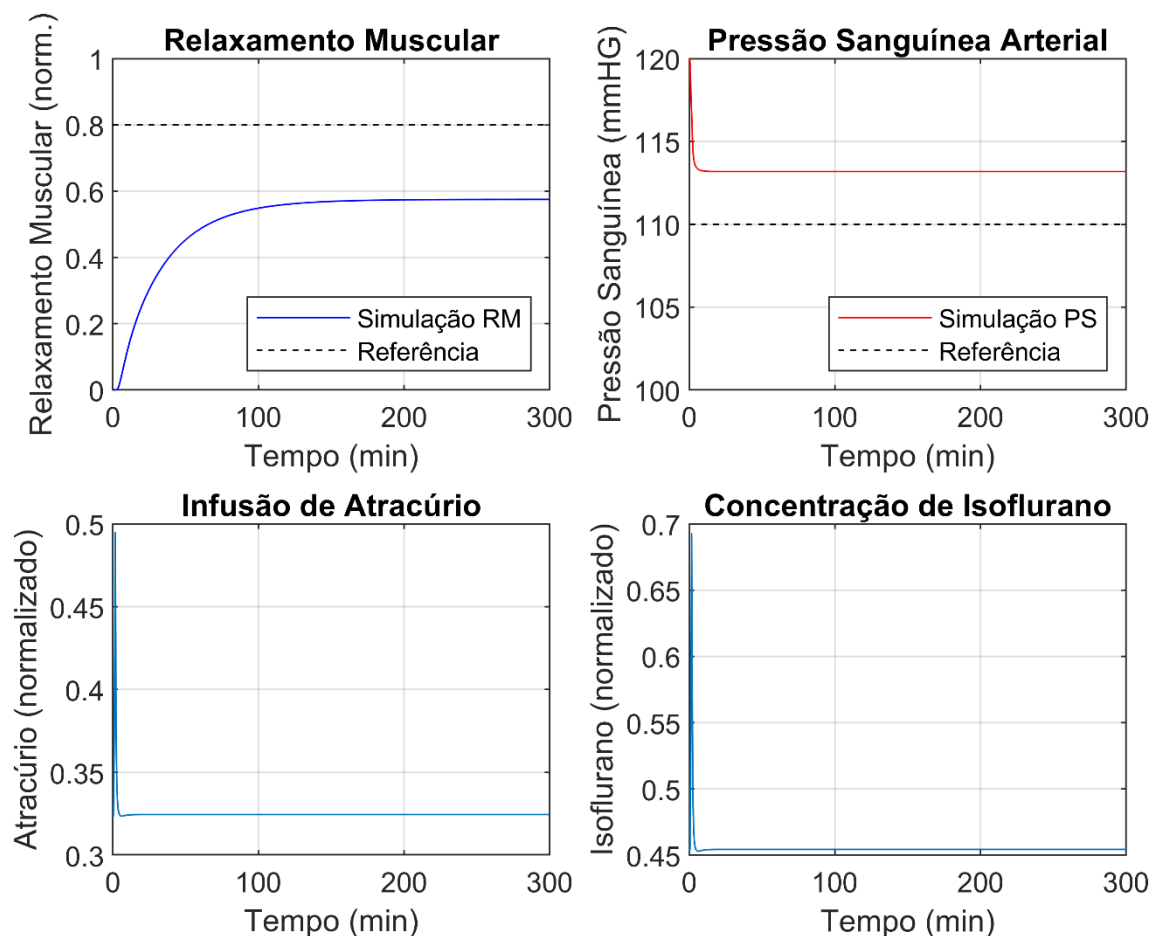
controlador é atingir tal referência. O tempo máximo significa o tempo em que a resposta do controlador foi simulada.

Considerando a Equação 24 na seção 2.3.4.6 Modelo Multivariável do Corpo Humano para Anestesia referente à Pressão Sanguínea (PS), quanto mais for adicionado o remédio, menor será a PS. Dessa forma, a simulação adiciona um valor de 120 mmHG que será considerado como o ponto de partida para a introdução do fármaco diminuir o valor até a referência. O Relaxamento Muscular não precisa de nenhum valor extra porque é considerado que o ponto de partida seja zero, ou seja, inicialmente o paciente não tem nenhum bloqueio neuromuscular.

4.1.1 Simulação do controlador T1.22.FLC

A simulação do controlador utilizou um *script* para calcular as respostas de duas saídas em malha fechada. O objetivo do controlador é atuar para que a saída se iguale a uma referência, ou seja, um valor específico que é escolhido anteriormente. O resultado da simulação está na Figura 26.

Figura 26 – Resultado da simulação do controlador T1.22.FLC



Fonte: Elaborada pelo Autor (2021)

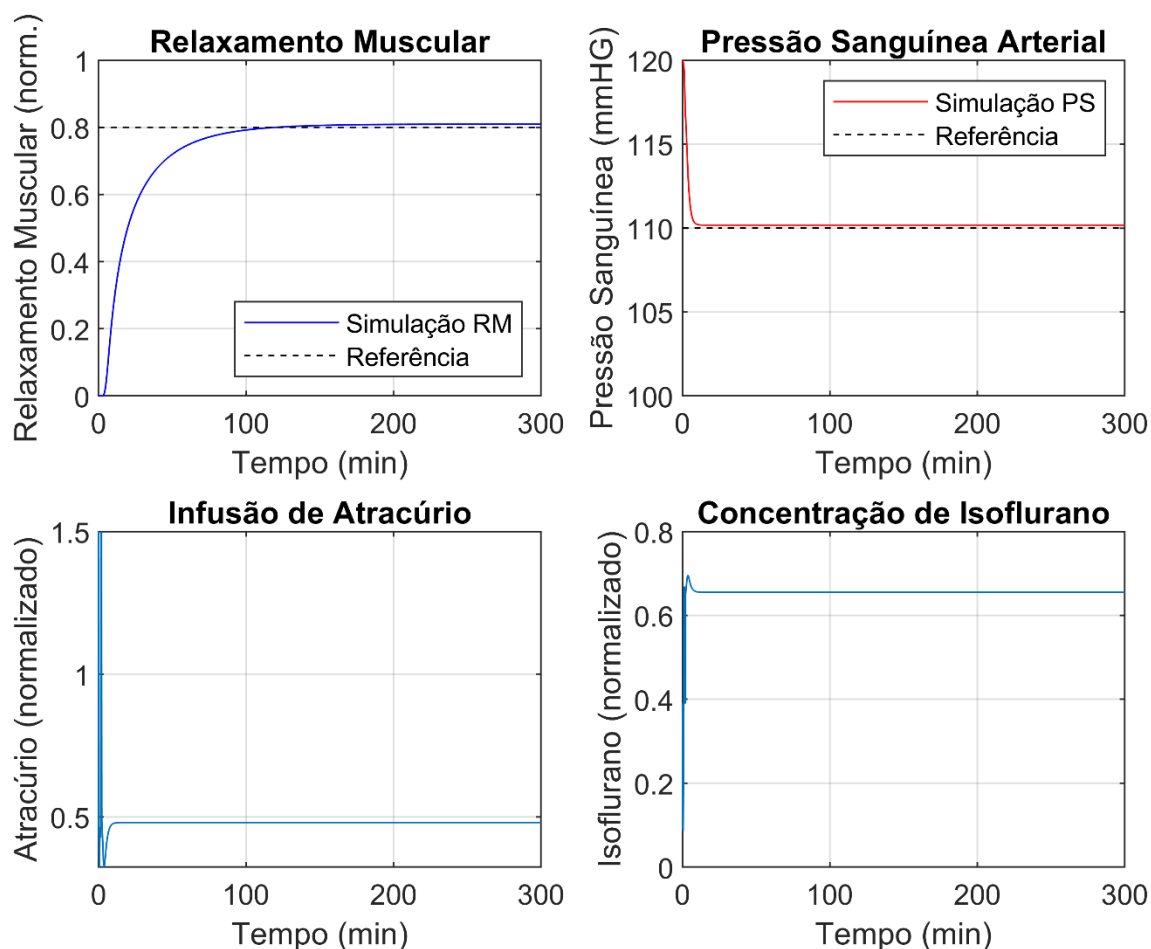
A Figura 26 apresenta o resultado da simulação do T1.22.FLC. São apresentadas quatro janelas que demonstram duas saídas e utilização de fármacos em um período de 300 minutos. A janela superior à esquerda mostra a simulação da saída Relaxamento Muscular normalizada em azul com a Referência de RM em preto com linha pontilhada. Na próxima janela superior à direita, tem-se a resposta da simulação da Pressão Sanguínea Arterial com sua unidade de mmHG e linha em vermelho e a Referência PS em preto pontilhado. As últimas duas janelas inferiores representam a utilização do remédio Atracúrio e Isoflurano em unidades normalizadas para a obtenção das respostas das janelas de cima. O comportamento das curvas em azul e em vermelho serão analisadas na seção 4.2 ANÁLISE DOS RESULTADOS, porém, o resultado foi condizente com o mostrado em outros trabalhos e o esperado pelos respectivos modelos descritos na seção 2.3.4.6 Modelo Multivariável do Corpo Humano para Anestesia. Basicamente, o

relaxamento muscular do paciente na simulação foi de 0 até aproximadamente 0,55 e a pressão sanguínea caiu de 120 mmHG para aproximadamente 116 mmHG.

4.1.2 Simulação do controlador T1.42.FLC

Similarmente à simulação anterior, o controlador T1.42.FLC utiliza a lógica *fuzzy* tipo-1 para levar as saídas até suas respectivas referências, com a diferença sendo duas entradas extras. Seu resultado da simulação se encontra na Figura 27.

Figura 27 – Resultado da simulação do controlador T1.42.FLC



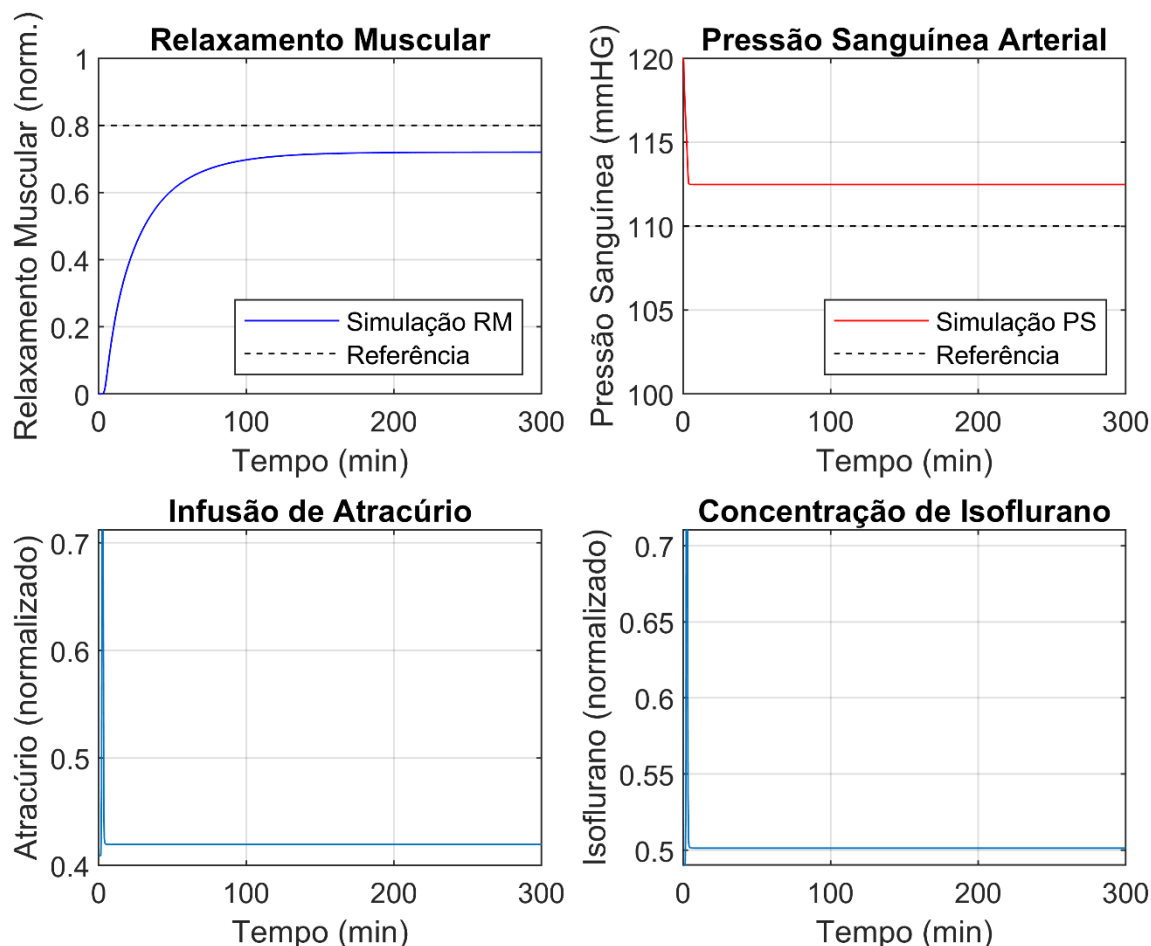
Fonte: Elaborada pelo Autor (2021)

A Figura 27 mostra o resultado da simulação do T1.42.FLC. De maneira similar à Figura 26, as quatro janelas representam os resultados das saídas Relaxamento Muscular e Pressão Sanguínea Arterial nas janelas superiores e utilização dos remédios Atracúrio e Isoflurano nas janelas inferiores com a mesma formatação utilizada anteriormente.

4.1.3 Simulação do controlador IT2.22.FLC

O controlador IT2.22.FLC utiliza a lógica *fuzzy* tipo-2 intervalar com duas entradas. Seu resultado da simulação se encontra na figura abaixo.

Figura 28 – Resultado da simulação do controlador IT2.22.FLC



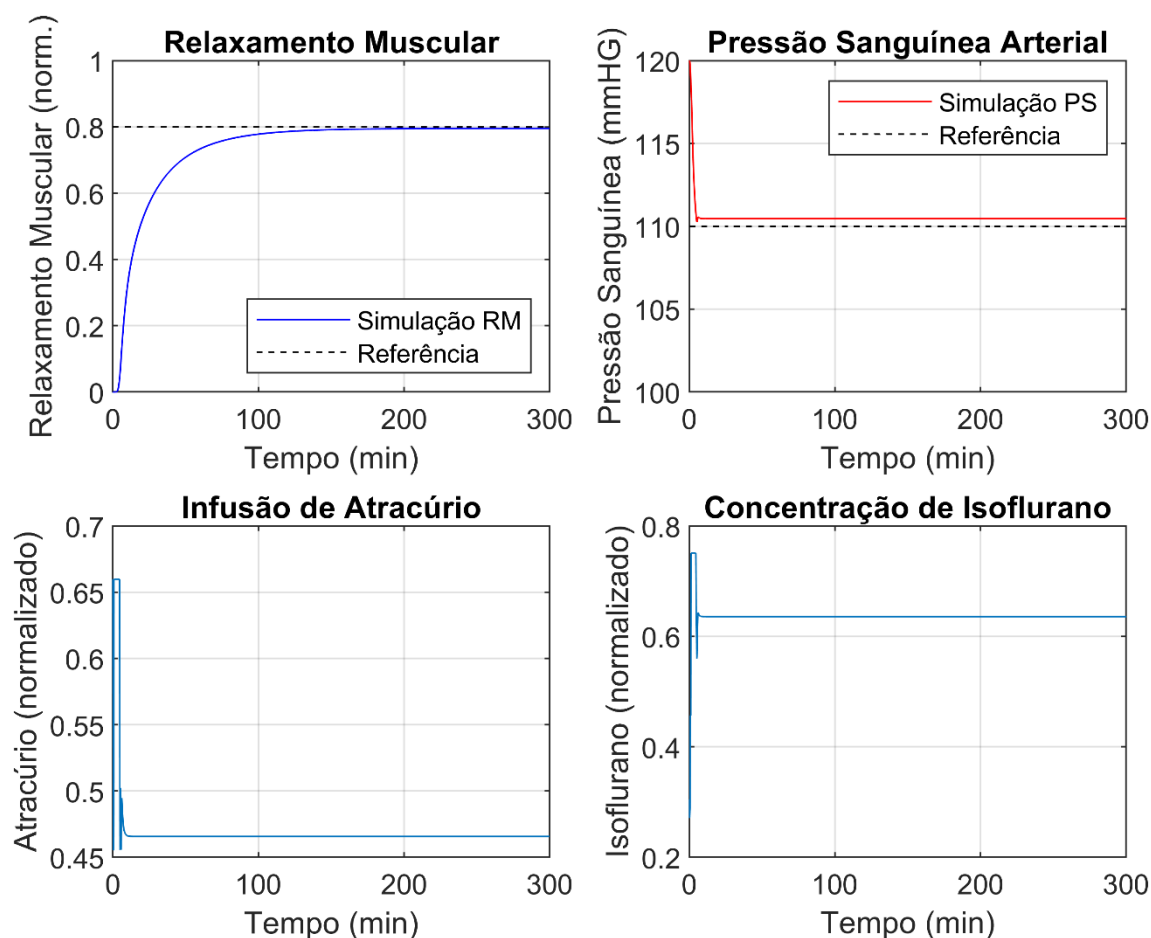
Fonte: Elaborada pelo Autor (2021)

O resultado da simulação do IT2.22.FLC (Figura 28) está disposto de maneira similar ao resultado da Figura 26 e Figura 27. Mais uma vez, as janelas superiores mostram as respostas das saídas enquanto as inferiores mostram as entradas.

4.1.4 Simulação do controlador IT2.42.FLC

A simulação final foi feita com o controlador IT2.42.FLC que utiliza quatro entradas de maneira similar ao T1.42.FLC. O resultado da simulação pode ser visto na figura abaixo.

Figura 29 – Resultado da simulação do controlador IT2.42.FLC



Fonte: Elaborada pelo Autor (2021)

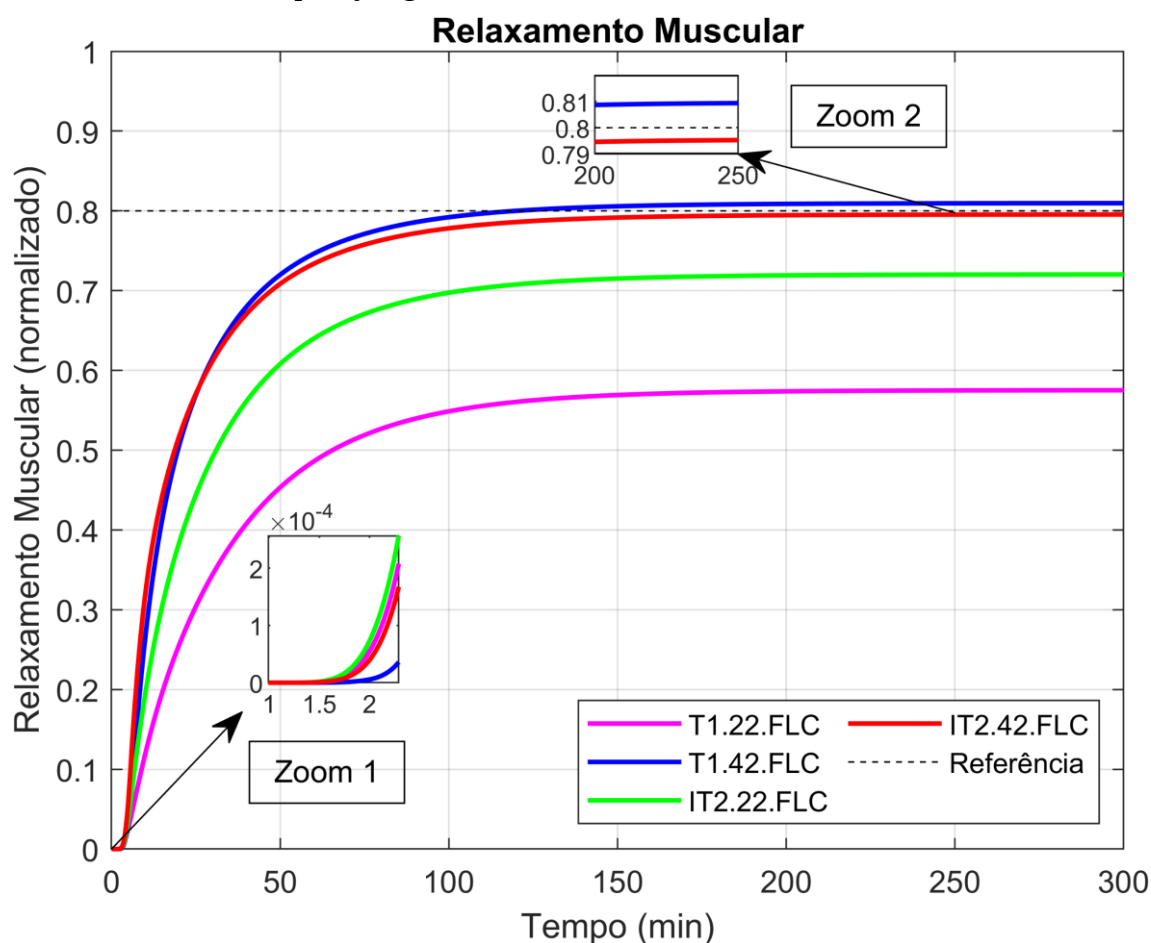
A Figura 29 mostra o resultado da simulação do IT2.42.FLC. Seguindo as mesmas especificações das figuras de resultados de simulação do T1.22.FLC, T1.42.FLC e IT2.22.FLC, demonstra o resultado da simulação em quatro janelas, duas delas para as saídas e as outras duas para as entradas.

4.1.5 Comparação das simulações

Todas as simulações foram feitas com parâmetros similares para uma comparação entre cada controlador. As simulações também ofereceram condições similares, utilizando o mesmo modelo com certas limitações discutidas na seção 2.3.4.6 Modelo Multivariável do Corpo Humano para Anestesia. A partir das respostas de cada controlador, foram compilados dois gráficos e duas tabelas de comparação direta em Relaxamento Muscular (Gráfico 1 e Tabela 9) e Pressão Sanguínea (Gráfico 2 e Tabela

10), e ainda foi incluído uma classificação em cada tabela para facilitar uma comparação mais precisa.

Gráfico 1 – Comparação gráfica entre os resultados em Relaxamento Muscular



Fonte: Elaborada pelo Autor (2021)

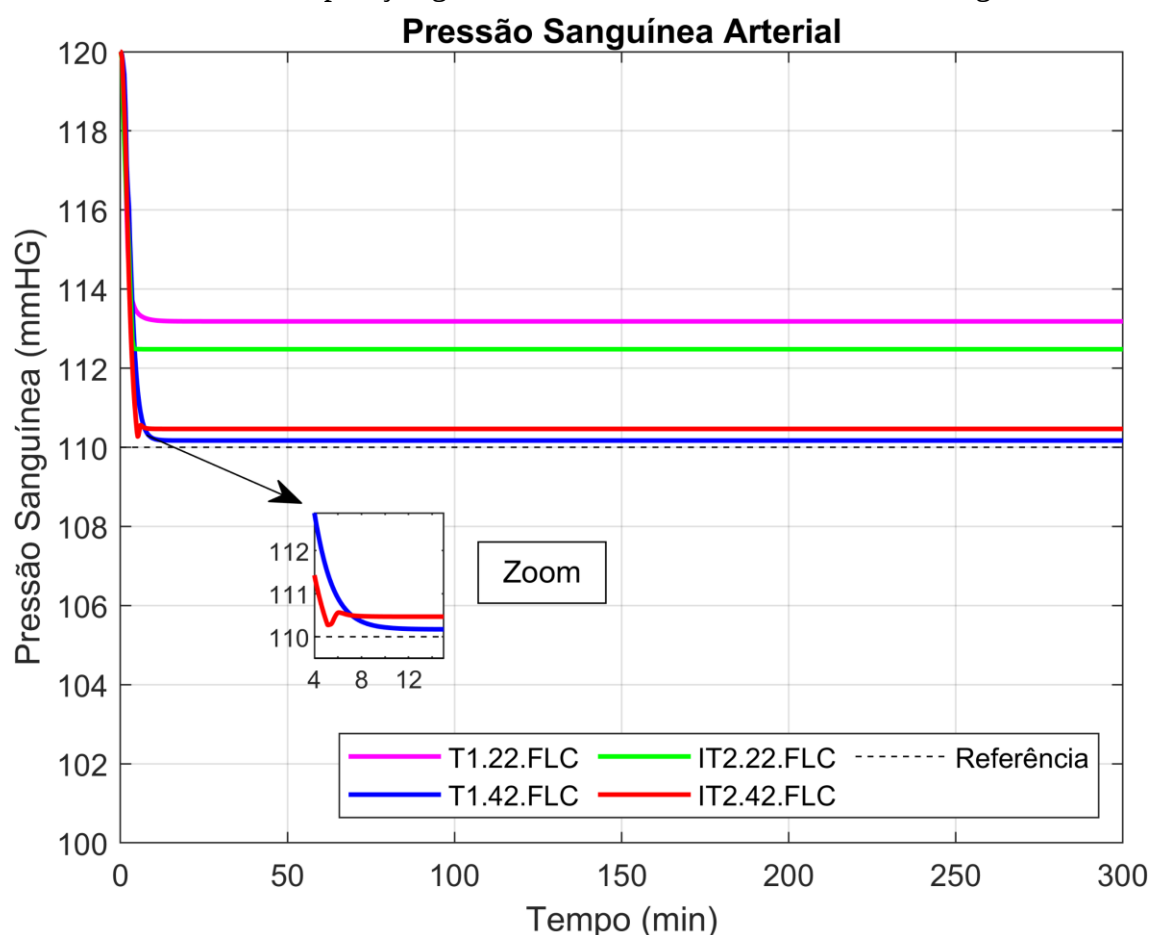
O Gráfico 1 mostra a comparação gráfica das respostas dos quatro controladores na saída Relaxamento Muscular, sendo uma compilação das janelas com RM da Figura 26, Figura 27, Figura 28 e Figura 29. Todos os controladores: T1.22.FLC (em magenta), T1.42.FLC (em azul), IT2.22.FLC (em verde) e IT2.42.FLC (em vermelho) têm a mesma Referência (em preto pontilhado). No gráfico há também duas janelas extras que ampliam uma parte específica, essas chamadas de Zoom 1 e Zoom 2. No Zoom 1, a seta está começando da origem em 1 minuto e terminando na janela que mostra uma ampliação da área inicial de todas as respostas. Enquanto o Zoom 2 amplia uma parte final em 200 minutos das respostas de T1.42.FLC e IT2.42.FLC.

Tabela 9 – Comparação numérica entre os resultados em Relaxamento Muscular

Parâmetro	T1.22.FLC	T1.42.FLC	IT2.22.FLC	IT2.42.FLC
Referência	0,8	0,8	0,8	0,8
Valor final	0,5751	0,8096	0,7203	0,7954
Módulo do erro	0,2249	0,0096	0,0797	0,0046
Classificação	4°	2°	3°	1°

Fonte: Elaborada pelo Autor (2021)

Na Tabela 9, encontra-se a comparação numérica dos resultados da saída Relaxamento Muscular dos controladores T1.22.FLC, T1.42.FLC, IT2.22.FLC e IT2.42.FLC nos parâmetros Referência que é a mesma para todos, Valor Final, Módulo do erro e Classificação.

Gráfico 2 – Comparação gráfica entre os resultados em Pressão Sanguínea

Fonte: Elaborada pelo Autor (2021)

O Gráfico 2 mostra a comparação gráfica das respostas dos quatro controladores na saída Pressão Sanguínea Arterial, baseando-se nos dados das janelas com PS da Figura 26, Figura 27, Figura 28 e Figura 29. Tais controladores são: T1.22.FLC (em magenta), T1.42.FLC (em azul), IT2.22.FLC (em verde) e IT2.42.FLC (em vermelho) e todos possuem a mesma Referência (em preto pontilhado). No gráfico foi incluída uma janela de ampliação chamada Zoom. Ela mostra uma seção logo após o começo em 4 minutos que evidencia uma mudança das respostas dos controladores T1.42.FLC e IT2.42.FLC.

Tabela 10 – Comparação numérica entre os resultados em Pressão Sanguínea

Parâmetro	T1.22.FLC	T1.42.FLC	IT2.22.FLC	IT2.42.FLC
Referência	110	110	110	110
Valor final	113,1841	110,1712	112,4803	110,4656
Módulo do erro	3,1841	0,1712	2,4803	0,4656
Classificação	4°	1°	3°	2°

Fonte: Elaborada pelo Autor (2021)

Na Tabela 10, tem-se a comparação numérica das respostas dos controladores em Pressão Sanguínea, disposta de maneira similar à Tabela 9.

4.2 ANÁLISE DOS RESULTADOS

Para a simulação do controlador T1.22.FLC (Figura 26), o relaxamento muscular e pressão sanguínea tiveram respostas com grande diferença ao valor ideal, logo seu desempenho foi baixo. Esse é o controlador mais simples em relação aos outros e não atingiu um resultado satisfatório, porém, mostrou estabilidade na sua resposta. As curvas das saídas Infusão de Atracúrio (Atr.inf) e Concentração de Isoflurano (Iso.conc) no controlador T1.22.FLC mostram um pico nos primeiros minutos com uma estabilização em um valor menor, sendo esses não suficientes para atingir o resultado desejado.

Já o T1.42.FLC teve o resultado de sua simulação (Figura 27) excelente porque o Relaxamento Muscular (RM) e a Pressão Sanguínea (PS) chegaram muito próximos às suas respectivas referências. Esse controlador é relativamente mais complexo com quatro entradas, em vez de duas do T1.22.FLC e possui mais regras. A resposta da entrada de Atr.inf foi similar ao T1.22.FLC, porque no começo houve um pico de indução e depois estabilizou para um valor fixo, com a diferença em que o valor de pico e estabilização foram maiores, assim, chegando mais próximo da resposta. Enquanto a resposta do Iso.conc foi mais estável com um valor controlado. Isso levou a uma melhor resposta da

PS, pois essa saída só depende do Isoflurano, de acordo com a Equação 24 na seção 2.3.4.6 Modelo Multivariável do Corpo Humano para Anestesia e Figura 12. Nesse caso, o maior número de entradas levou a um melhor desempenho, isso pode ser explicado também pelas novas entradas serem integrais dos erros, assim tendo informações extras de como as saídas estão reagindo às entradas.

O controlador IT2.22.FLC mostrou um desempenho mediano de sua simulação (Figura 28) em suas saídas por conta de uma diferença ou erro considerável em relação às referências de cada saída. Esse utiliza a lógica *fuzzy* tipo-2 intervalar e é mais robusto às incertezas em sua estrutura. Comparando o IT2.22.FLC diretamente ao T1.22.FLC, sendo que os dois possuem duas entradas e utilizam as mesmas bases de regras, o controlador com lógica *fuzzy* tipo-2 intervalar tem um melhor desempenho em RM e PS, mas não consegue se sair melhor do que o T1.42.FLC. As respostas das induções dos fármacos são bem similares ao T1.22.FLC, e também não foram suficientes, sustentando a ideia de que a base de regras não está completa para um bom controle das variáveis.

Enquanto a resposta do IT2.42.FLC (Figura 29) teve um desempenho excelente em Relaxamento Muscular (RM), mas com um erro pequeno aparente em Pressão Sanguínea (PS), assim não sendo melhor que T1.42.FLC em PS e vencendo todos os outros controladores. Os fármacos tiveram uma indução menos estáveis em relação às outras simulações, sendo um indício de que seus escalares ainda poderiam ser ajustados para um melhor desempenho, considerando a indução estável e suave do T1.42.FLC, que possui o mesmo número de entradas e base de regras.

O Relaxamento Muscular apresenta em seu um modelo um atraso de transporte de um minuto, representado por um exponencial elevado a um s negativo multiplicado por 1, como mostrado na Equação 23 e Equação 25 na seção 2.3.4.6 Modelo Multivariável do Corpo Humano para Anestesia, ou seja, o efeito demora esse tempo a partir da entrada do fármaco, e ainda é considerada a parte não linear utilizando a equação de Hill, isso torna o sistema complexo e difícil de controlar. A comparação no Gráfico 1 mostra em sua janela Zoom 1 o atraso de transporte. Também demonstra que os melhores desempenhos foram do T1.42.FLC em azul e IT2.42.FLC em vermelho. O pior desempenho foi do T1.22.FLC, enquanto o IT2.22.FLC foi melhor do que sua versão com lógica *fuzzy* tipo-1, mas não conseguiu atingir perto o suficiente do objetivo para ser considerado bom. Outra comparação mais precisa na Tabela 9 com os valores exatos das respostas em estado estacionário, ou seja, após o seu período de grande variação, irá

demonstrar qual controlador teve o melhor desempenho, que nesse caso se resume ao menor erro em relação à referência.

A Tabela 9 mostra os parâmetros de comparação entre as respostas dos controladores ao Relaxamento Muscular (RM). Todos tiveram as mesmas referências com valores finais diferentes, calculando assim o erro ou a diferença entre os dois. O menor módulo do erro mostra qual classificação cada controlador teve e nesse caso o IT2.42.FLC se saiu um pouco melhor em relação ao T1.42.FLC em segundo lugar. Por fim, os controladores T1.22.FLC e IT2.22.FLC tiveram suas classificações previsíveis pela comparação gráfica no Gráfico 1. A janela de Zoom 2 da figura apoia o resultado final, podendo ver claramente que a linha vermelha (IT2.42.FLC) está mais próxima da referência do que a linha azul (T1.42.FLC).

O modelo da Pressão Sanguínea apresenta um atraso de transporte de 0,42 minuto na Equação 24, ou seja, a equação possui um exponencial elevado a um s negativo e multiplicado por 0,42, porém, diferentemente do Relaxamento Muscular, não apresenta parte não linear. A comparação no Gráfico 2 mostra que o melhor desempenho foi do T1.42.FLC em azul e o IT2.42.FLC em vermelho ficou logo atrás com uma resposta menos suave, como destacado na janela de Zoom do gráfico. O pior desempenho foi do T1.22.FLC, enquanto o IT2.22.FLC foi um pouco melhor. Uma comparação com valores mostrados na Tabela 10 das respostas em estado estacionário irá dizer qual o resultado final de cada controlador, usando os mesmos critérios da Tabela 9.

A Tabela 10 mostrou o desempenho dos controladores no controle da Pressão Sanguínea. O método de comparação é o mesmo da Tabela 9 e os resultados são também similares, porém, o T1.42.FLC em primeiro lugar teve um erro menor do que IT2.42.FLC, ficando em segundo lugar. O resto da classificação é igual ao RM.

Existem limitações nas condições do escopo deste trabalho com relação às simulações, por exemplo, não é considerado variação inter paciente e intra paciente e não é adicionado nenhum tipo de incerteza de medição (ruído). Logo, os resultados só dizem a respeito nas condições de modelagem com parâmetros fixos e sem ruídos.

Considerando toda a análise dos resultados, este trabalho referenciado pela proposta de aplicação da técnica de controle *fuzzy* tipo-2 em anestesia geral teve resultado satisfatório com relação às técnicas de controle *fuzzy* tipo-1. Percebe-se claramente que o desempenho é superior ao tipo-1 e um maior número de entradas é benéfico.

4.3 PROTÓTIPO REPRESENTATIVO DO SISTEMA

O sistema real de controle da anestesia possuiria sistemas complexos biomédicos com confiabilidade suficiente para serem utilizados em ambiente hospitalar. Uma representação do sistema físico por meio de um protótipo evidenciará os principais equipamentos necessários. O Quadro 8 mostra a comparação entre o equipamento que deveria ser utilizado na aplicação real e o equipamento que foi utilizado para representar o sistema.

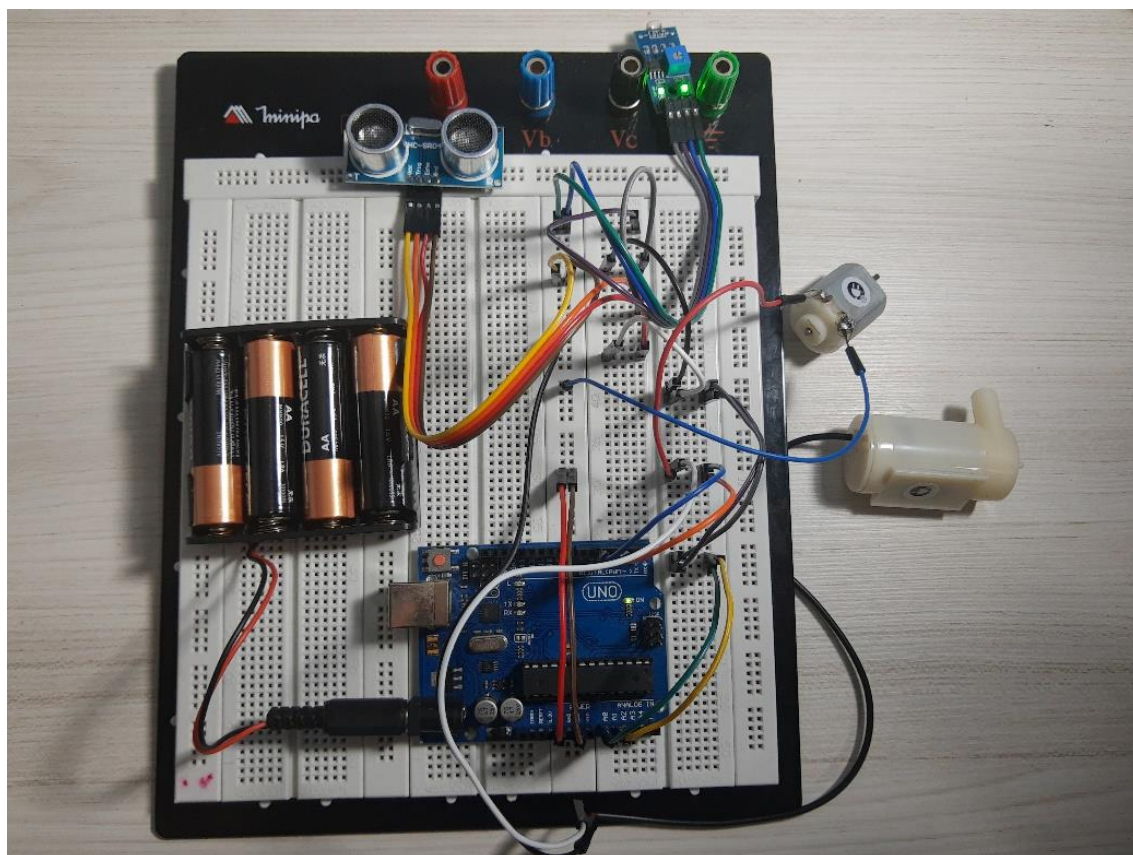
Quadro 8 – Comparação da utilização dos equipamentos no protótipo

Equipamento real	Representação
Computador para o processamento de dados dos sensores, algoritmo do controlador <i>fuzzy</i> e controle dos atuadores	Arduino UNO R3
Sensor para o relaxamento muscular ou eletromiografia (EMG)	Módulo Sensor de Luz LDR
Sensor para a pressão sanguínea	Sensor Ultrassônico HC-SR04
Ventilador para a administração do isoflurano	Mini Motor DC 3V a 6V
Bomba de seringa para a administração do atracúrio	Mini Bomba Submersível DC 3V – 6V 120L

Fonte: Elaborado pelo Autor (2021)

A plataforma embarcada Arduino serve como um computador capaz de receber os sinais dos sensores, executar o algoritmo de controle *fuzzy* e enviar sinais elétricos para os atuadores, não levando em consideração a capacidade, desempenho ou confiabilidade necessária para o uso real. A escolha dos sensores representativos se deu por sua saída analógica, levando em consideração que os sensores reais também teriam esse tipo de sinal. Já os atuadores da emulação têm uma certa semelhança direta com seus pares, ou seja, motor DC para ventilador e bomba submersível para bomba de seringa, porém, se diferem em tamanho, potência e especialidade.

A montagem da emulação do sistema é demonstrada na Figura 30, sendo que foram utilizados alguns equipamentos extras na montagem que estão listados na Quadro 1 na coluna de *hardware*.

Figura 30 – Protótipo do sistema montado

Fonte: Elaborada pelo Autor (2021)

A montagem mostrada na figura acima representa o controle da anestesia. O protoboard foi utilizado para permitir as conexões entre os equipamentos eletrônicos, mas em um ambiente profissional poderia ter sido utilizado uma placa de circuito impresso (PCB) com a mesma função.

O Arduino é alimentado por uma fonte externa composta por quatro pilhas AA (4 x 1,5V = 6V). Suas entradas utilizadas são A0 para o sensor Ultrassônico e A1 para o Sensor LDR, ambas têm a capacidade de processar entradas analógicas pelo conversor analógico digital (ADC) com 10 bits de resolução para uma tensão de referência de 5V, ou seja, é possível ler uma entrada com até 5V com 1024 níveis de tensão, sendo que na programação isso será representado por um número de 0 a 1023, enquanto as saídas estão localizadas nos pinos digitais D3 com o Motor DC e D5 com a Bomba DC. Essas saídas têm a capacidade de modulação por largura de pulso (PWM) com valores possíveis de 0 a 255 que definem a tensão de saída de 0 a 5V (ORGANTINI, 2018).

O protótipo construído que representou fisicamente o sistema real para o controle da anestesia serviu como uma visualização de tal controle, também evidenciou as

restrições com relação à qualidade, desempenho e confiabilidade dos equipamentos que deveriam ser utilizados caso fosse construído uma máquina biomédica.

5 CONSIDERAÇÕES FINAIS

Para a realização deste trabalho foi feito um estudo sobre lógica e controle *fuzzy* tipo-1 e tipo-2, definição e procedimentos da anestesia geral, erros de dosagem e modelagem farmacocinética farmacodinâmica e, por último, controle automático da anestesia. Além disso, foram projetados controladores *fuzzy* e construídas simulações e um protótipo de pequena escala. Uma análise dos resultados e conclusão foram as últimas realizadas no trabalho.

Quanto aos objetivos do trabalho, foram estudados vários conceitos da lógica e controle *fuzzy* e sua aplicação e teoria referente a anestesia geral no capítulo 2. Ainda nesse capítulo, os problemas existentes em caso de erro de dosagem na anestesia foram evidenciados junto com a frequência de ocorrência de erros em geral na anestesia. As técnicas de controle *fuzzy* tipo-1 e tipo-2 também foram identificados no capítulo 2, enquanto seus projetos foram realizados seguindo as diretrizes descritas no capítulo 3, nas seções 3.3 PROJETO DOS CONTROLADORES *FUZZY* TIPO-1 (T1.FLC) e 3.4 PROJETO DE CONTROLADORES *FUZZY* TIPO-2 INTERVALAR (IT2.FLC). Os resultados finais das simulações foram verificados e comparados no capítulo 4, nas seções 4.1 SIMULAÇÕES e 4.2 ANÁLISE DOS RESULTADOS.

A análise das simulações dos quatro controladores evidencia um desempenho superior na utilização de controladores com quatro entradas (T1.42.FLC e IT2.42.FLC) em relação aos controladores que utilizam somente duas entradas (T1.22.FLC e IT2.22.FLC). A lógica *fuzzy* tipo-2 intervalar utilizada nos controladores IT2.22.FLC e IT2.42.FLC resultou em um melhor desempenho em relação aos controladores *fuzzy* tipo-1 (T1.22.FLC e T1.42.FLC) de uma maneira em que para dois controladores com o mesmo número de entradas, por exemplo T1.22.FLC e IT2.22.FLC, o controlador utilizando lógica *fuzzy* tipo-2 intervalar teve o melhor resultado final em relação ao que utiliza lógica *fuzzy* tipo-1 na maioria dos casos.

Foram encontrados vários obstáculos no decorrer do trabalho, esses sendo escassez de artigos científicos na língua portuguesa sobre o tema específico, complexidade de ler e compreender artigos já escritos sobre controle *fuzzy* tipo-2 de anestesia geral e dificuldade de escrever códigos de simulação.

Dessa forma, a utilização do controle *fuzzy* tipo-2 é uma opção viável para um sistema automático de anestesia geral e uma boa alternativa para controlar sistemas dinâmicos não lineares com múltiplas entradas e múltiplas saídas.

5.1 TRABALHOS FUTUROS

Devido à grande variedade de temas de pesquisa similares ao descrito no trabalho, algumas sugestões para realizar trabalhos futuros são:

- a) Utilizar uma estrutura de controlador *fuzzy* tipo-2 intervalar autoajustável (SOFLC) no controle de anestesia geral utilizando programação genética (GP) para adquirir uma base de regras mais robusta e adaptável e comparar com os outros controladores *fuzzy*.
- b) Implementar um controlador neuro-*fuzzy* tipo-2 intervalar no controle de anestesia geral medindo o Índice Bispectral (BIS) para a profundidade da anestesia (DoA).
- c) Realizar um estudo clínico, com o devido consentimento e ética, do controle automático de anestesia geral utilizando o controlador *fuzzy* tipo-2 intervalar.

REFERÊNCIAS

- AGUIRRE, L. A. *et al.* (ed.). **Enciclopédia de Automática: controle e automação**. São Paulo: Blucher, 2007. 3 v. (3).
- ANNIE, S. J.; THIRILOGASUNDARY, M. R.; KUMAR, V. R. H. Drug administration errors among anesthesiologists: The burden in India—A questionnaire-based survey. **Journal of Anaesthesiology, Clinical Pharmacology**, v. 35, n. 2, p. 220, 2019.
- AMIRI, A. A. *et al.* Comparação entre anestesia intravenosa e inalatória na náusea e vômito pós-operatórios em laparotomia: estudo clínico randomizado. **Revista Brasileira de Anestesiologia**, v. 70, p. 471-476, 2020.
- ALEXANDER, J. C.; ROMITO, B. T.; ÇOBANOĞLU, M. C. The present and future role of artificial intelligence and machine learning in anesthesiology. **International anesthesiology clinics**, v. 58, n. 4, p. 7-16, 2020.
- AZAM, M. H. *et al.* Fuzzy type-1 triangular membership function approximation using fuzzy C-means. In: **2020 International Conference on Computational Intelligence (ICCI)**. IEEE, 2020. p. 115-120.
- BERNARDES, R. **Sistemas Neuro-Fuzzy Aplicados no Controle de uma aeronave de asa fixa**. 2019. 67f. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecatrônica) – Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, Uberlândia.
- CHEN, C. *et al.* A comprehensive study of the efficiency of type-reduction algorithms. **IEEE Transactions on Fuzzy Systems**, v. 29, n. 6, p. 1556-1566, 2020.
- COPOT, D. (ed.). **Automated drug delivery in anesthesia**. London: Academic Press, 2020. 338 p.
- DHAWAN, I. *et al.* Erros de medicação em anestesia: inaceitável ou inevitável?. **Revista Brasileira de Anestesiologia**, v. 67, p. 184-192, 2017.
- DOCTOR, F. *et al.* Type-2 fuzzy sets applied to multivariable self-organizing fuzzy logic controllers for regulating anesthesia. **Applied Soft Computing**, v. 38, p. 872-889, 2016.

ERDMANN, T. R. *et al.* Perfil de erros de administração de medicamentos em anestesia entre anesthesiologistas catarinenses. **Revista Brasileira de Anesthesiologia**, v. 66, p. 105-110, 2016.

FARIAS, M. S. **Realimentação de estados baseada em regras fuzzy tipo-2 para servocontrole de sistemas não lineares**. 2019. 148 p. Tese (Doutorado em Engenharia Elétrica e de Computação) – Centro de Tecnologia, Universidade Federal do Rio Grande do Norte, Natal, 2019.

GARG, R. *et al.* Automatic Anesthesia Injector. **Asian Journal For Convergence In Technology (AJCT)**, v. 5, n. 2, 2019.

GOMIDE, F.; GUDWIN, R. R.; TANSCHKEIT, R. Conceitos fundamentais da teoria de conjuntos fuzzy, lógica fuzzy e aplicações. **Proc. 6 th IFSA Congress-Tutorials**. p. 1-38, 1995.

GUILLET, V. M. M. **Análise de fornecedores de um setor público empregando o método Fuzzy-TOPSIS com auxílio do método AHP**. 2019. 92 p. Dissertação (Mestrado em Engenharia de Produção) – Centro de Tecnologia, Universidade Federal de Santa Maria, Santa Maria, 2019.

ILYAS, M. *et al.* Controle da hipnose na administração de propofol com base na estratégia de controle não linear. **Revista Brasileira de Anesthesiologia**, v. 67, p. 122-130, 2017.

KHALIFA, T. R. *et al.* A novel fuzzy Wiener-based nonlinear modelling for engineering applications. **ISA transactions**, v. 97, p. 130-142, 2020.

KHODAEI, M. J. *et al.* An adaptive multi-critic neuro-fuzzy control framework for intravenous anesthesia administration. **IFAC-PapersOnLine**, v. 51, n. 34, p. 202-207, 2019.

MENDEL, J. M. **Uncertain rule-based fuzzy systems: Introduction and new directions**. 2nd ed. Cham: Springer, 2017. 684 p.

MÉNDEZ, J. A. *et al.* Adaptive fuzzy predictive controller for anesthesia delivery. **Control Engineering Practice**, v. 46, p. 1-9, 2016.

MÉNDEZ, J. A. *et al.* Improving the anesthetic process by a fuzzy rule based medical decision system. **Artificial intelligence in medicine**, v. 84, p. 159-170, 2018.

NASCIMENTO JÚNIOR, C. L.; YONEYAMA, T. **Inteligência Artificial: em controle e automação**. São Paulo: Blucher, 2000.

ORGANTINI, G. Arduino as a tool for physics experiments. In: **Journal of Physics: Conference Series**. IOP Publishing, v. 1076, n. 1, p. 012026, 2018.

RIZOL, P.; MESQUITA, L.; SAOTOME, O. Lógica fuzzy tipo-2. **Revista SODEBRAS**, v. 6, p. 27-46, 2011.

SANDRI, S.; CORREA, C. Lógica nebulosa. **V Escola de Redes Neurais**, v. 1, n. 1, p. 73-90, 1999.

SADATI, N.; HOSSEINZADEH, M.; DUMONT, G. A. Multi-model robust control of depth of hypnosis. **Biomedical Signal Processing and Control**, v. 40, p. 443-453, 2018.

SILVEIRA, D. T.; CÓRDOVA, F. P. A pesquisa científica. In: GERHART T. E.; SILVEIRA, D. T. (org.). **Métodos de pesquisa**. Porto Alegre: Editora da UFRGS, 2009. cap. 2, p. 31-42.

SHARMA, R. *et al.* An optimal interval type-2 fuzzy logic control based closed-loop drug administration to regulate the mean arterial blood pressure. **Computer methods and programs in biomedicine**, v. 185, p. 105167, 2020.

SHIEH, J. S. *et al.* Monitoring and control of anesthesia using multivariable self-organizing fuzzy logic structure. In: JIN Y.; WANG L. (ed.). **Fuzzy Systems in Bioinformatics and Computational Biology**. Berlin: Springer, Berlin, Heidelberg, 2009. cap. 14, p. 273-295.

SIMÕES, M. G.; SHAW, I. S. **Controle e Modelagem Fuzzy**. 2. ed. São Paulo: Blucher, 2007.

SIVANANDAM, S. N.; SUMATHI S.; DEEPA S. N. **Introduction to fuzzy logic using MATLAB**. Berlin Heidelberg: Springer, 2007. 430 p.

SOUZA, D.; RIZOL, P. M. S. R. Projeto de um circuito de máximo cmos para uso em um controlador fuzzy tipo-2 intervalar. **Revista SODEBRAS**, v. 12, n. 140, p. 103-107, 2017.

- TAHERIYAN, F.; GHAFOURIAN, M. S.; NOORI, A. Applying GA optimization algorithm for interval type-2 Fuzzy logic controller parameters of multivariable anesthesia system. *In: Iranian Conference on Electrical Engineering (ICEE)*, 26., Mashhad. IEEE. p. 1613-1618, 2018.
- VASCONCELLOS, B. S. **Proposta de procedimento para projeto de controladores fuzzy multivariáveis**. 2017. 109 p. Dissertação (Mestrado em Engenharia Elétrica e de Computação) – Centro de Tecnologia, Universidade Federal do Rio Grande do Norte, Natal, 2017.
- WEI, Z. X. *et al.* An optimized type-2 self-organizing fuzzy logic controller applied in anesthesia for propofol dosing to regulate BIS. **IEEE Transactions on Fuzzy Systems**, v. 28, n. 6, p. 1062-1072, 2020.
- YAO, D.; WANG, C. Interval type-2 fuzzy information measures and their applications to attribute decision-making approach. **Journal of Intelligent & Fuzzy Systems**, v. 33, n. 3, p. 1809-1821, 2017.

APÊNDICE A – PROJETO DO CONTROLADOR T1.22.FLC

Função para gerar o controlador Fuzzy Tipo 1 (T1_22_FLC)

Autor: Giovanni Oliveira de Sousa

Ano: 2021

2 ENTRADAS: RM_E, PS_E

2 SAÍDAS: ATRAC_INF.M, ISO_CONC.M

```
clc
clear
close all
% O controlador é baseado em SHIEH et al. (2009)
% https://doi.org/10.1007/978-3-540-89968-6_14
% O controlador possui 2 entradas (inputs) e 1 saída (outputs);
% O controlador será decomposto em 2 controladores simples;
% Um controlador simples será dedicado para uma saída;
% Logo, cada controlador simples possui 2 entradas e 1 saída;
% Cada entrada possui 7 funções de pertinência (MF);
% Cada saída possui 4 funções de pertinência (MF);
% Os controladores serão declarados sem regras;
T1_22_FLC_Anest(1:2) = mamfis; % Vetor nulo de controladores fuzzy
Nome_ctr = ["Controlador 1 T1FLC", "Controlador 2 T1FLC"]; % Nome de cada controlador
for i = 1:2
    T1_22_FLC_Anest(i) =
mamfis("Name", Nome_ctr(i), 'NumInputs', 2, 'NumInputMFs', 7, ...
'NumOutputs', 1, 'NumOutputMFs', 4, 'AddRules', "none");
end
```

1. DEFINIR PARÂMETROS FIXOS:

A. NOME (STRING) DAS ENTRADAS (INPUTS) E SAÍDAS (INPUTS):

Entradas: Relaxamento Muscular erro (**RM_e**); Pressão Sanguínea erro (**PS_e**).

Saídas: Mudança na Infusão de Atracúrio (**Atrac_Inf.M**); Mudança na Concentração de Isoflurano (**Iso_Conc.M**)

B. NOME (STRING) DAS FUNÇÕES DE PERTINÊNCIA (MFS) DAS ENTRADAS E SAÍDAS:

Entradas: Negativo Grande (**NB**); Negativo Médio (**NM**); Negativo Pequeno (**NS**); Zero (**ZE**); Positivo Pequeno (**PS**); Positivo Médio (**PM**); Positivo Grande (**PB**).

Saídas: Zero (**ZE**); Positivo Pequeno (**PS**); Positivo Médio (**PM**); Positivo Grande (**PB**).

C. VALORES DOS CONJUNTOS FUZZY (FUNÇÕES DE PERTINÊNCIA - MF) NOMINAIS TIPO 1:

Os conjuntos nominais são MFs triangulares com largura de 4 e sobreposição de 25%:

NS - [-4 -2 0];

ZE - [-2 0 2];

PS - [0 2 4];

...

```
% Entradas
Nome_inputs = ["RM.e","PS.e"];

NMFs_inputs = [ "NB" , "NM" , "NS" , "ZE" , "PS" , "PM"
, "PB" ];
Val_MFInputs = [ -8 -6 -4 ; -6 -4 -2 ; -4 -2 0 ; -2 0 2 ; 0 2 4 ; 2 4 6
; 4 6 8 ];

% Saídas
Nome_outputs = ["Atr.inf","Iso.conc"];

NMFs_outputs = [ "ZE" , "PS" , "PM" , "PB" ];
Val_MFOutputs = [ -2 0 2 ; 0 2 4 ; 2 4 6 ; 4 6 8 ];
```

2. DEFINIR OS PARÂMETROS DE CADA MF DA I/O

A. DEFINIR OS NOMES (STRINGS) DE CADA ENTRADA E SAÍDA;

B. DEFINIR A FAIXA DE VALORES (UNIVERSO DE DISCURSO) DE CADA ENTRADA E SAÍDA;

C. DEFINIR CADA FUNÇÃO DE PERTINÊNCIA (MF) COMO TIPO TRIANGULAR (TRIMF);

Aqui as funções de pertinência (MFs) são definidas como tipo triangular (*trimf*) com seus parâmetros fixos. A faixa de valores está entre -6 e 6, porém, só serve para as entradas que admitem valores negativos. As saídas só podem aceitar valores positivos, logo sua faixa está entre 0 e 6.

```
% Entradas: 2 entradas com 7 MFs
for i = 1:2 % Controladores para todas as saídas
    for j = 1:2 % Entradas para cada controlador
        T1_22_FLC_Anest(i).Inputs(j).Name = Nome_inputs(j);
        T1_22_FLC_Anest(i).Inputs(j).Range = [-6 6];
        for k = 1:7 % Funções de Pertinência por entrada em cada
controlador
            T1_22_FLC_Anest(i).Inputs(j).MembershipFunctions(k).Name =
NMFs_inputs(k);
```

```

        T1_22_FLC_Anest(i).Inputs(j).MembershipFunctions(k).Type =
"trimf";

T1_22_FLC_Anest(i).Inputs(j).MembershipFunctions(k).Parameters =
Val_MFInputs(k,:);
        end
    end
end

% Saídas: 1 saída com 4 MFs
for i = 1:2 % Controladores para todas as saídas
    T1_22_FLC_Anest(i).Outputs(1).Name = Nome_outputs(i);
    T1_22_FLC_Anest(i).Outputs(1).Range = [0 6];
    for j = 1:4 % Funções de Pertinência por saída em cada controlador
        T1_22_FLC_Anest(i).Outputs(1).MembershipFunctions(j).Name =
NMFs_outputs(j);
        T1_22_FLC_Anest(i).Outputs(1).MembershipFunctions(j).Type =
"trimf";
        T1_22_FLC_Anest(i).Outputs(1).MembershipFunctions(j).Parameters
= Val_MFOutputs(j,:);
    end
end

```

3. DEFINIR A BASE DE REGRAS

As regras já foram definidas na referência de acordo com as imagens abaixo. Para definir no programa é necessário criar uma matriz com o tamanho correto de combinações possíveis. Cada regra terá o formato de:

SE Entrada (1,2) == (NB - 1 ... PB - 7) **E** Entrada (1,2) == (NB - 1 ... PB - 7) **ENTÃO** Saída (1,2) == (ZE - 1 ... PB - 4)

Cada regra pode ser descrita somente com os índices de entrada(s), função de pertinência(s) e saída(s).

Cada controlador terá uma base de regras distinta, ou seja, para cada saída há uma base de regras.

Atra_Inf	NB	NM	NS	ZE	PS	PM	PB
NB	PB		PS		ZE		ZE
NM		PM		ZE		ZE	
NS	PB		PS		ZE		ZE
ZE		PM		ZE		ZE	
PS	PM		ZE		ZE		ZE
PM		PM		ZE		ZE	
PB	PM		ZE		ZE		ZE

Fonte: SHIEH et al. (2009)

Iso_Conc	NB	NM	NS	ZE	PS	PM	PB
NB	ZE		ZE		ZE		ZE
NM		ZE		ZE		ZE	
NS	ZE		ZE		ZE		ZE
ZE		ZE		ZE		ZE	
PS	PS		PS		PS		ZE
PM		PM		PS		ZE	
PB	PB		PM		PS		PS

Fonte: SHIEH et al. (2009)

```

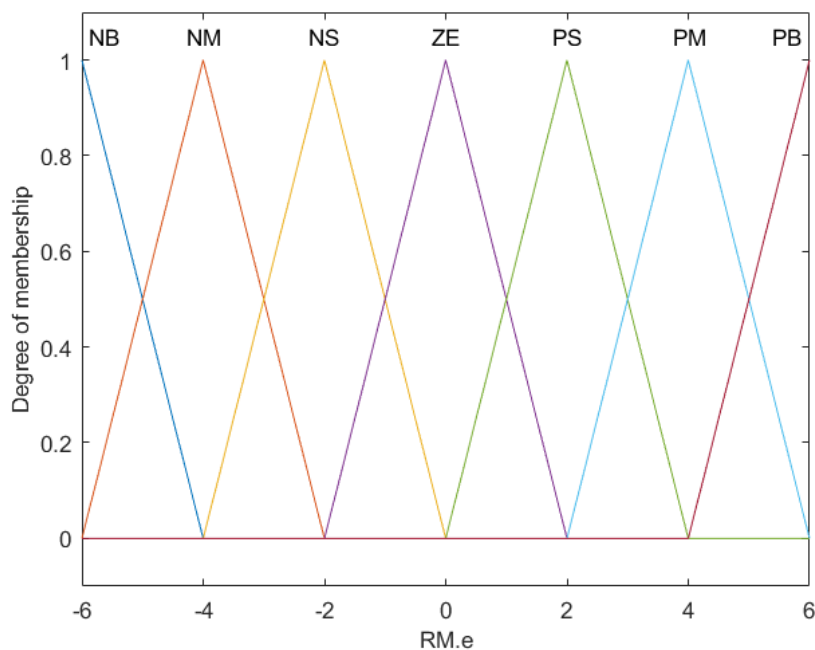
Regras_Fuzzy_T1_22 = zeros(25,5*2);
Regras_Fuzzy_T1_22(:,4:5) = 1;
Regras_Fuzzy_T1_22(:,9:10) = 1;
O1 = [4 2 1 1 , 3 1 1 , 4 2 1 1 , 3 1 1 , 3 1 1 1 , 3 1 1 , 3 1 1 1];
O2 = [1 1 1 1 , 1 1 1 , 1 1 1 1 , 1 1 1 , 2 2 2 1 , 3 2 1 , 4 3 2 2];
O = [O1;O2];
cont1 = 1;
cont2 = 1;
for i = 1:25
    for j = 1:2
        d = 5*(j-1);
        Regras_Fuzzy_T1_22(i,1+d) = cont1;
        Regras_Fuzzy_T1_22(i,2+d) = cont2;
        Regras_Fuzzy_T1_22(i,3+d) = O(j,i);
    end
end

```

```

end
if cont1+2 <= 7
    cont1 = cont1 + 2;
else
    if cont1+2 == 8
        cont1 = 1;
    else
        cont1 = 2;
    end
    cont2 = cont2 + 1;
end
end
for i = 1:2
    d = 5*(i-1);
    T1_22_FLC_Anest(i) =
addRule(T1_22_FLC_Anest(i),Regras_Fuzzy_T1_22(1:25,1+d:5+d));
end
Fuzzifi_MFs(1:2) = figure;
Fuzzifi_MFs(1) = figure('Name','Fuzzificação RM.e','NumberTitle','off');
plotmf(T1_22_FLC_Anest(1),'input',1);
exportgraphics(Fuzzifi_MFs(1),'Fuzzificacao_RM_e.tif','Resolution',600);

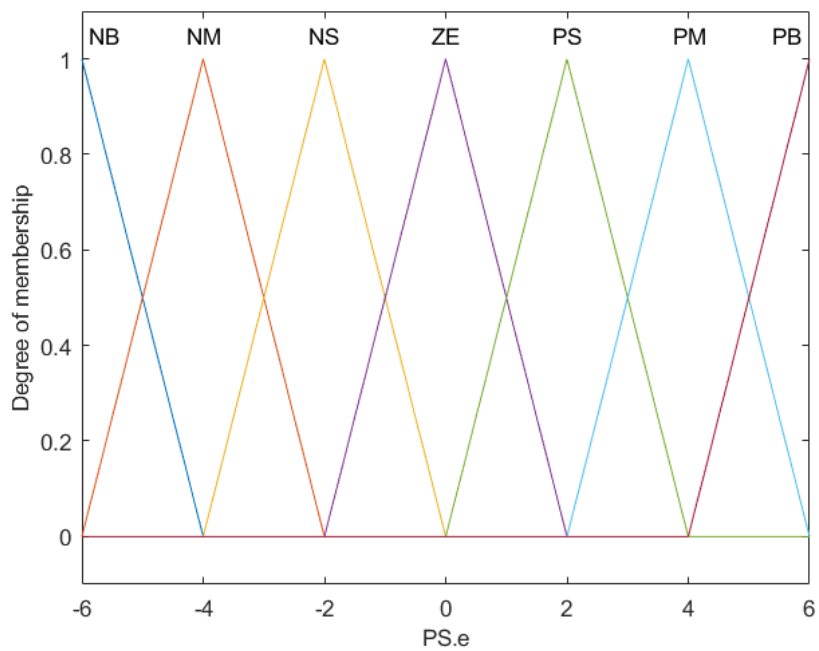
```



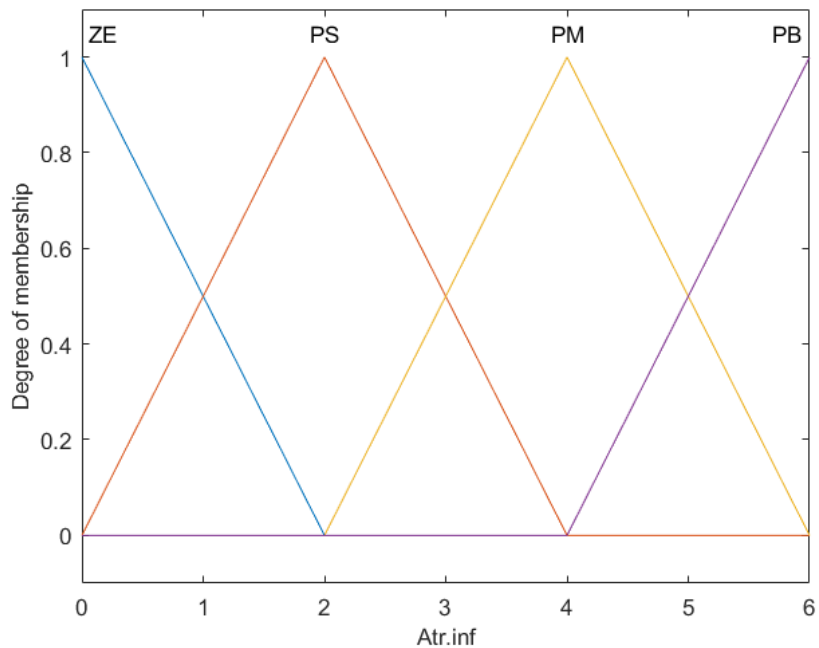
```

Fuzzifi_MFs(2) = figure('Name','Fuzzificação PS.e','NumberTitle','off');
plotmf(T1_22_FLC_Anest(2),'input',2);
exportgraphics(Fuzzifi_MFs(2),'Fuzzificacao_PS_e.tif','Resolution',600);

```

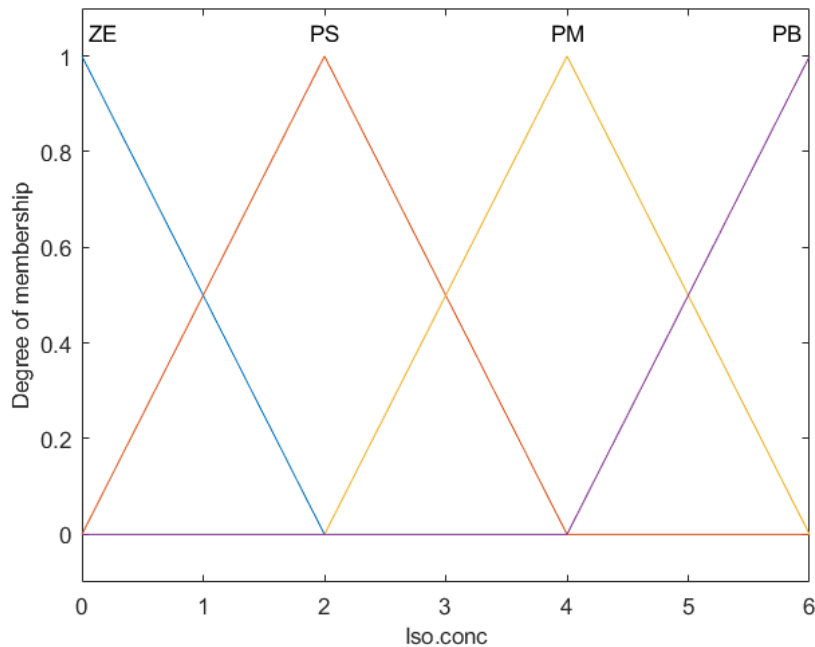


```
Defuzzi_MFs(1:2) = figure;
Defuzzi_MFs(1) = figure('Name','Defuzzificação
Atr.inf','NumberTitle','off');
plotmf(T1_22_FLC_Anest(1),'output',1);
exportgraphics(Defuzzi_MFs(1),'Defuzzificacao_Atr_inf.tif','Resolution',
600);
```



```
Defuzzi_MFs(2) = figure('Name','Defuzzificação
Iso.conc','NumberTitle','off');
```

```
plotmf(T1_22_FLC_Anest(2), 'output', 1);
exportgraphics(Defuzzi_MFs(2), 'Defuzzificacao_Iso_conc.tif', 'Resolution', 600);
```



```
disp(T1_22_FLC_Anest(1));
```

mamfis with properties:

```

        Name: "Controlador 1 T1FLC"
        AndMethod: "min"
        OrMethod: "max"
        ImplicationMethod: "min"
        AggregationMethod: "max"
        DefuzzificationMethod: "centroid"
        Inputs: [1x2 fisvar]
        Outputs: [1x1 fisvar]
        Rules: [1x25 fisrule]
        DisableStructuralChecks: 0

```

See 'getTunableSettings' method for parameter optimization.

```
disp(T1_22_FLC_Anest(2));
```

mamfis with properties:

```

        Name: "Controlador 2 T1FLC"
        AndMethod: "min"
        OrMethod: "max"

```

```
ImplicationMethod: "min"  
AggregationMethod: "max"  
DefuzzificationMethod: "centroid"  
    Inputs: [1x2 fisvar]  
    Outputs: [1x1 fisvar]  
    Rules: [1x25 fisrule]  
DisableStructuralChecks: 0
```

See 'getTunableSettings' method for parameter optimization.

```
save T1_22_FLC.mat T1_22_FLC_Anest;
```

APÊNDICE B – PROJETO DO CONTROLADOR T1.42.FLC

Função para gerar o controlador Fuzzy tipo 1 (T1_42_FLC)

Autor: Giovanni Oliveira de Sousa

Ano: 2021

4 ENTRADAS: RM_E, RM_INT.E, PS_E, PS_INT.E

2 SAÍDAS: ATRAC_INF.M, ISO_CONC.M

```
clc
clear
close all
% O controlador é baseado em SHIEH et al. (2009)
% https://doi.org/10.1007/978-3-540-89968-6_14
% O controlador possui 4 entradas (inputs) e 2 saídas (outputs);
% O controlador será decomposto em 12 controladores simples;
% Seis controladores simples são dedicados para cada saída;
% Logo, cada controlador simples possui 2 entradas e 1 saída;
% Cada entrada possui 7 funções de pertinência (MF);
% Cada saída possui 4 funções de pertinência (MF);
% O controlador será declarado sem regras;
T1_42_FLC_Anest(1:2,1:6) = mamfis; % Vetor nulo de controladores fuzzy
% Nome de cada controlador
Nome_ctr = ["Controlador 11 T1FLC","Controlador 21 T1FLC","Controlador 31 T1FLC",...
            "Controlador 41 T1FLC","Controlador 51 T1FLC","Controlador 61 T1FLC";...
            "Controlador 12 T1FLC","Controlador 22 T1FLC","Controlador 32 T1FLC",...
            "Controlador 42 T1FLC","Controlador 52 T1FLC","Controlador 62 T1FLC"];
for i = 1:6
    for j = 1:2
        T1_42_FLC_Anest(j,i) =
mamfis("Name",Nome_ctr(j,i),'NumInputs',2,'NumInputMFs',7,...
        'NumOutputs',1,'NumOutputMFs',4,'AddRules',"none");
    end
end
end
```

1. DEFINIR PARÂMETROS FIXOS:

A. NOME (STRING) DAS ENTRADAS (INPUTS) E SAÍDAS (INPUTS):

Entradas: Relaxamento Muscular erro (**RM_e**); Relaxamento Muscular integral erro (**RM_int.e**); Pressão Sanguínea erro (**PS_e**); Pressão Sanguínea integral erro (**PS_int.e**).

Saídas: Mudança na Infusão de Atracúrio (**Atrac_Inf.M**); Mudança na Concentração de Isoflurano (**Iso_Conc.M**)

B. NOME (STRING) DAS FUNÇÕES DE PERTINÊNCIA (MFS) DAS ENTRADAS E SAÍDAS:

Entradas: Negativo Grande (**NB**); Negativo Médio (**NM**); Negativo Pequeno (**NS**); Zero (**ZE**); Positivo Pequeno (**PS**); Positivo Médio (**PM**); Positivo Grande (**PB**).

Saídas: Zero (**ZE**); Positivo Pequeno (**PS**); Positivo Médio (**PM**); Positivo Grande (**PB**).

C. VALORES DOS CONJUNTOS FUZZY (FUNÇÕES DE PERTINÊNCIA - MF) NOMINAIS TIPO 1:

Os conjuntos nominais são MFs triangulares com largura de 4 e sobreposição de 25%:

NS - [-4 -2 0];

ZE - [-2 0 2];

PS - [0 2 4];

...

```
% Entradas
Nome_inputs = ["RM_e","RM_int.e","PS_e","PS_int.e"];

Ordem_inputs = [1 1 1 , 2 2 3 ; 2 3 4 , 3 4 4]; % Ordem das entradas

NMFs_inputs = [ "NB" , "NM" , "NS" , "ZE" , "PS" , "PM" , "PB" ];
Val_MFInputs = [ -8 -6 -4 ; -6 -4 -2 ; -4 -2 0 ; -2 0 2 ; 0 2 4 ; 2 4 6 ; 4 6 8 ];

% Saídas
Nome_outputs = ["Atr_Inf.m","Iso.Conc_m"];

NMFs_outputs = [ "ZE" , "PS" , "PM" , "PB" ];
Val_MFOutputs = [ -2 0 2 ; 0 2 4 ; 2 4 6 ; 4 6 8 ];
```

2. DEFINIR OS PARÂMETROS DE CADA MF DA I/O

A. DEFINIR OS NOMES (STRINGS) DE CADA ENTRADA E SAÍDA;

B. DEFINIR A FAIXA DE VALORES (UNIVERSO DE DISCURSO) DE CADA ENTRADA E SAÍDA;

C. DEFINIR CADA FUNÇÃO DE PERTINÊNCIA (MF) COMO TIPO TRIANGULAR (TRIMF);

Aqui as funções de pertinência (MFs) são definidas como tipo triangular (*trimf*) com seus parâmetros fixos. A faixa de valores está entre -6 e 6, porém, só serve para as entradas que admitem valores negativos. As saídas só podem aceitar valores positivos, logo sua faixa está entre 0 e 6.

```
% Entradas: 2 entradas com 7 MFs
for i = 1:6 % Controladores por saída
    for j = 1:2 % Saídas totais
        for k = 1:2 % Entradas em cada controlador
            T1_42_FLC_Anest(j,i).Inputs(k).Name =
Nome_inputs(Ordem_inputs(k,i));
            T1_42_FLC_Anest(j,i).Inputs(k).Range = [-6 6];
            for p = 1:7 % Funções de Pertinência por entrada em cada
controlador

T1_42_FLC_Anest(j,i).Inputs(k).MembershipFunctions(p).Name =
NMFs_inputs(p);

T1_42_FLC_Anest(j,i).Inputs(k).MembershipFunctions(p).Type = "trimf";
            T1_42_FLC_Anest(j,i).Inputs(k).MembershipFunctions(p)...
                .Parameters = Val_MFInputs(p,:);
            end
        end
    end
end

% Saídas: 1 saída com 4 MFs
for i = 1:6 % Controladores por saída
    for j = 1:2 % Saídas totais (uma saída por controlador)
        T1_42_FLC_Anest(j,i).Outputs(1).Name = Nome_outputs(j);
        T1_42_FLC_Anest(j,i).Outputs(1).Range = [0 6];
        for k = 1:4 % Funções de Pertinência por saída em cada
controlador

            T1_42_FLC_Anest(j,i).Outputs(1).MembershipFunctions(k).Name
= NMFs_outputs(k);
            T1_42_FLC_Anest(j,i).Outputs(1).MembershipFunctions(k).Type
= "trimf";
            T1_42_FLC_Anest(j,i).Outputs(1).MembershipFunctions(k)...
                .Parameters = Val_MFOutputs(k,:);
            end
        end
    end
end
```

3. DEFINIR A BASE DE REGRAS

As regras já foram definidas na referência de acordo com as imagens abaixo. Para definir no programa é necessário criar uma matriz com o tamanho correto de combinações possíveis. Cada regra terá o formato de:

SE Entrada (1,2,3,4) == (NB - 1 ... PB - 7) **E** Entrada (1,2,3,4) == (NB - 1 ... PB - 7) **ENTÃO**
 Saída (1,2) == (ZE - 1 ... PB - 4)

Cada regra pode ser descrita somente com os índices de entrada(s), função de pertinência(s) e saída(s).

Cada controlador terá uma base de regras distinta, ou seja, para cada saída há seis bases de regras.

Atra_Inf	NB	NM	NS	ZE	PS	PM	PB
NB	ZE PB PB PB		ZE PM PS PS PM PS		ZE ZE ZE ZE		ZE ZE ZE ZE
NM		ZE PM PM PM PM PM		ZE ZE ZE ZE		ZE ZE ZE ZE	
NS	ZE PM PB PB		ZE PS PS PS		ZE ZE ZE ZE		ZE ZE ZE ZE
ZE		ZE PM PM PM PM PM		ZE ZE ZE ZE		ZE ZE ZE ZE	
PS	ZE PS PM PM PB PB		ZE PS ZE ZE PS PS		ZE ZE ZE ZE		ZE ZE ZE ZE
PM		ZE PS PM PM PS PM		ZE ZE ZE ZE		ZE ZE ZE ZE	
PB	ZE PS PM PM PB PB		ZE PS ZE ZE PS PS		ZE ZE ZE ZE		ZE ZE ZE ZE

Fonte: SHIEH et al. (2009)

Iso_Conc	NB	NM	NS	ZE	PS	PM	PB
NB	ZE PB ZE ZE ZE ZE		ZE PM ZE ZE ZE ZE		ZE ZE ZE ZE		PS ZE ZE ZE ZE ZE
NM		ZE PM ZE ZE ZE ZE		ZE ZE ZE ZE		PS ZE ZE ZE	
NS	ZE PS ZE ZE ZE ZE		ZE PS ZE ZE ZE ZE		PS ZE ZE ZE		PM ZE ZE ZE ZE ZE
ZE		ZE ZE ZE ZE ZE ZE		ZE ZE ZE ZE		PM ZE ZE ZE	
PS	ZE ZE PS PS PS PM		PS ZE PS PS PS PS		PS ZE PS PS PS ZE		PB ZE ZE ZE ZE ZE
PM		PS ZE PM PM PM PM		PM ZE PS PS PS PS		PB ZE ZE PS	
PB	PS ZE PB PB PB PB		PM ZE PM PM PM PM		PB ZE PS PS PS PS		PB ZE PS PS PS PS

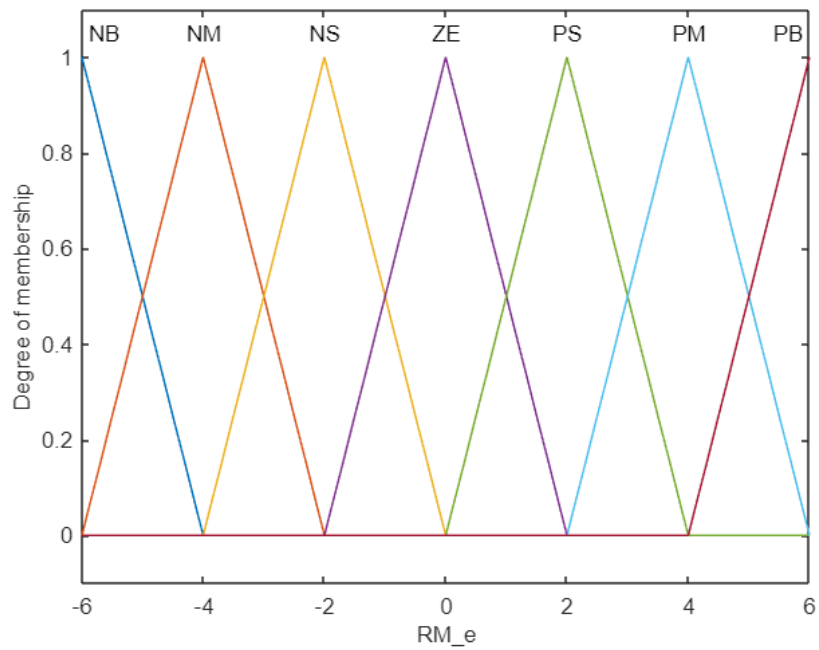
Fonte: SHIEH et al. (2009)

```

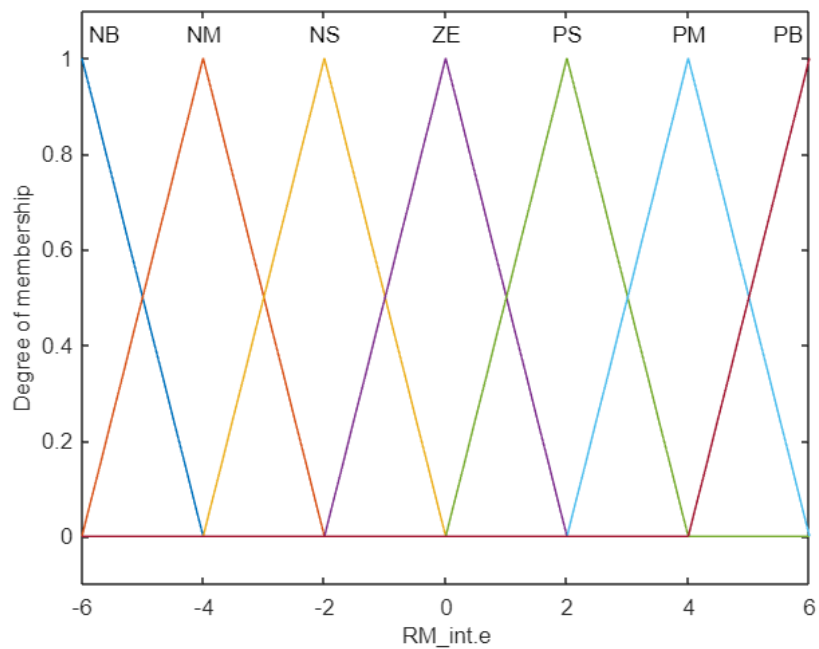
Regras_Fuzzy_T1_42 = zeros(6*25,5*2);
Regras_Fuzzy_T1_42(:,4:5) = 1;
Regras_Fuzzy_T1_42(:,9:10) = 1;
011 = [4 2 1 1 , 3 1 1 , 4 2 1 1 , 3 1 1 , 3 1 1 1 , 3 1 1 , 3 1 1 1];
012 = [4 3 1 1 , 3 1 1 , 4 2 1 1 , 3 1 1 , 4 2 1 1 , 2 1 1 , 4 2 1 1];
013 = [4 2 1 1 , 3 1 1 , 4 2 1 1 , 3 1 1 , 4 2 1 1 , 3 1 1 , 4 2 1 1];
014 = [4 2 1 1 , 3 1 1 , 4 2 1 1 , 3 1 1 , 3 1 1 1 , 3 1 1 , 3 1 1 1];
015 = [4 3 1 1 , 3 1 1 , 3 2 1 1 , 3 1 1 , 2 2 1 1 , 2 1 1 , 2 2 1 1];
016 = [1 1 1 1 , 1 1 1 , 1 1 1 1 , 1 1 1 , 1 1 1 1 , 1 1 1 , 1 1 1 1];
021 = [1 1 1 1 , 1 1 1 , 1 1 1 1 , 1 1 1 , 2 2 2 1 , 3 2 1 , 4 3 2 2];
022 = [1 1 1 1 , 1 1 1 , 1 1 1 1 , 1 1 1 , 2 2 2 1 , 3 2 1 , 4 3 2 2];
023 = [1 1 1 1 , 1 1 1 , 1 1 1 1 , 1 1 1 , 3 2 1 1 , 3 2 2 , 4 3 2 2];
024 = [1 1 1 1 , 1 1 1 , 1 1 1 1 , 1 1 1 , 2 2 2 1 , 3 2 1 , 4 3 2 2];
025 = [4 3 1 1 , 3 1 1 , 2 2 1 1 , 1 1 1 , 1 1 1 1 , 1 1 1 , 1 1 1 1];
026 = [1 1 1 2 , 1 1 2 , 1 1 2 3 , 1 1 3 , 1 2 2 4 , 2 3 4 , 2 3 4 4];
0 = [011,012,013,014,015,016;021,022,023,024,025,026];
cont1 = 1;
cont2 = 1;
for i = 1:6*25
    for j = 1:2
        d = 5*(j-1);
        Regras_Fuzzy_T1_42(i,1+d) = cont1;
        Regras_Fuzzy_T1_42(i,2+d) = cont2;
        Regras_Fuzzy_T1_42(i,3+d) = 0(j,i);
    end
    if cont1 == 7 && cont2 == 7
        cont1 = 1;
        cont2 = 1;
    end
    if cont1+2 <= 7
        cont1 = cont1 + 2;
    else
        if cont1+2 == 8
            cont1 = 1;
        else
            cont1 = 2;
        end
        cont2 = cont2 + 1;
    end
end
for i = 1:6
    for j = 1:2
        di1 = 25*(i-1) + 1;
        di2 = 25*i;
        dj1 = 5*(j-1) + 1;
        dj2 = 5*j;
        T1_42_FLC_Anest(j,i) = addRule(T1_42_FLC_Anest(j,i),...
            Regras_Fuzzy_T1_42(di1:di2,dj1:dj2));
    end
end

```

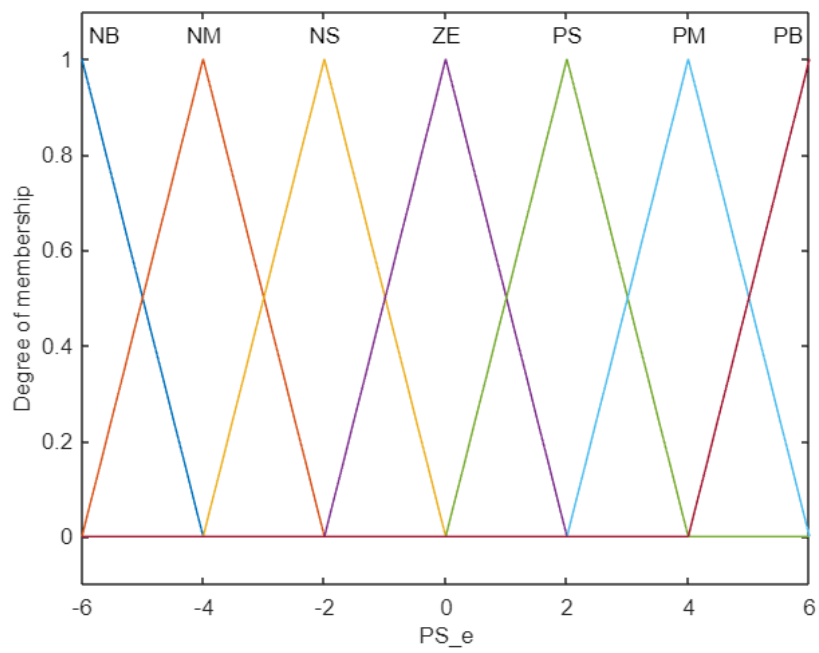
```
end
plotmf(T1_42_FLC_Anest(1,1), 'input', 1);
```



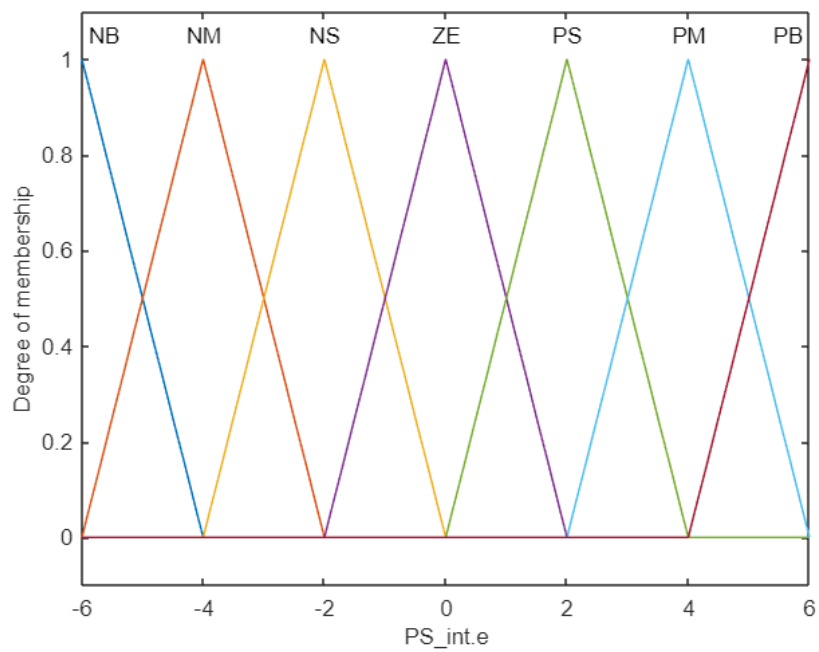
```
plotmf(T1_42_FLC_Anest(1,1), 'input', 2);
```



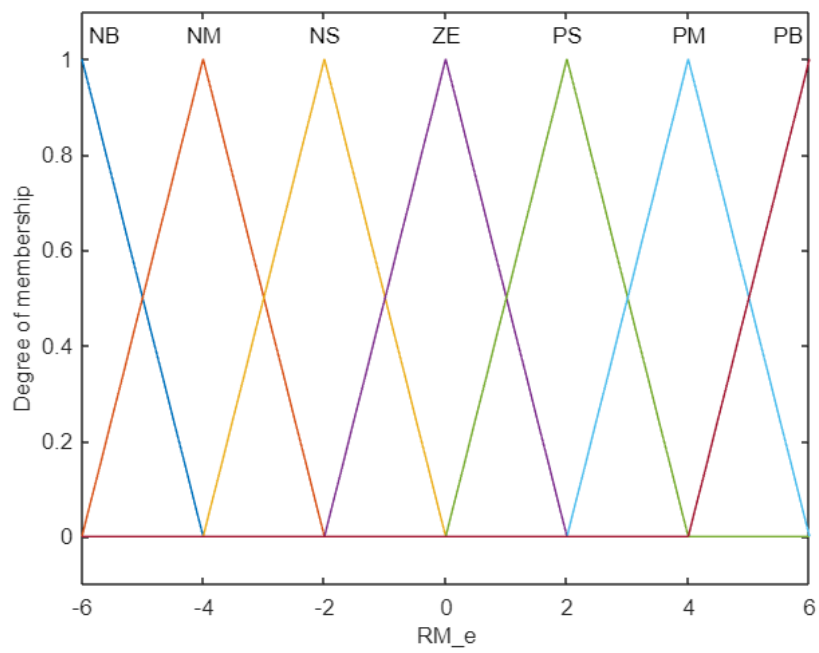
```
plotmf(T1_42_FLC_Anest(1,6), 'input', 1);
```



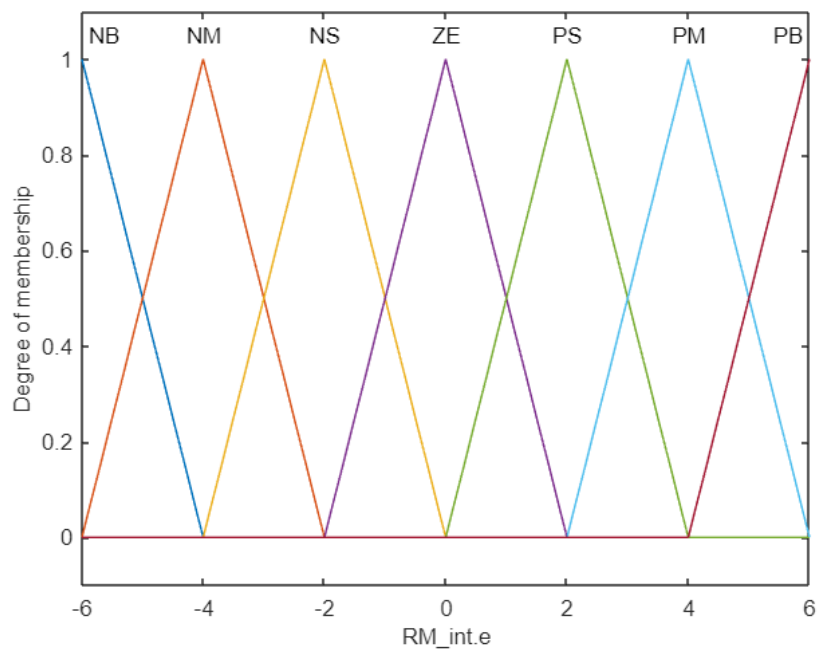
```
plotmf(T1_42_FLC_Anest(1,6), 'input', 2);
```



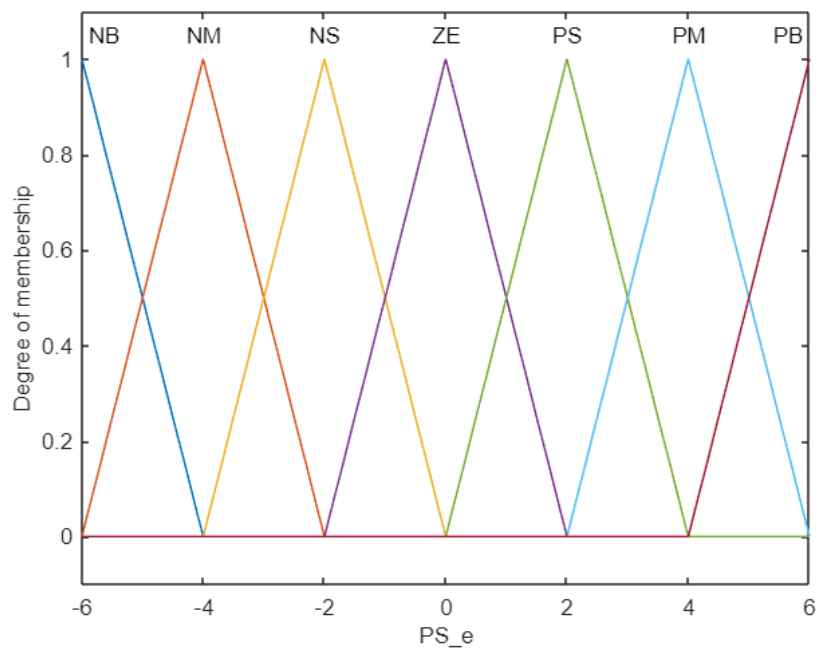
```
plotmf(T1_42_FLC_Anest(2,1), 'input', 1);
```



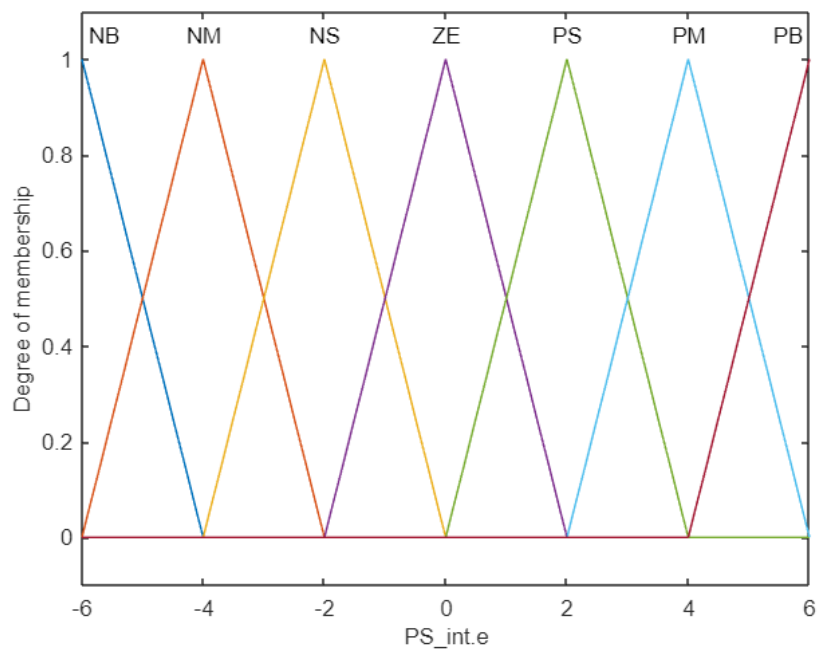
```
plotmf(T1_42_FLC_Anest(2,1), 'input', 2);
```



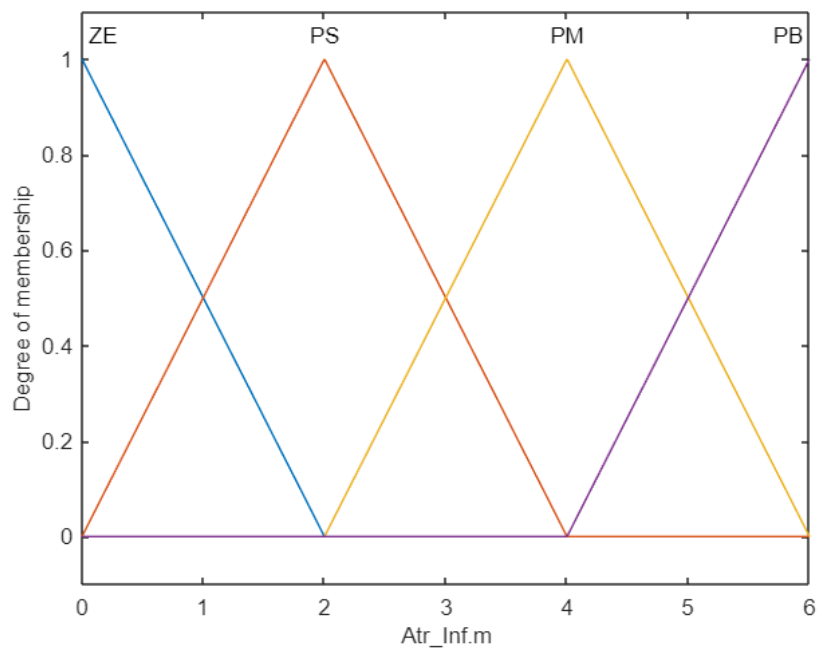
```
plotmf(T1_42_FLC_Anest(2,6), 'input', 1);
```



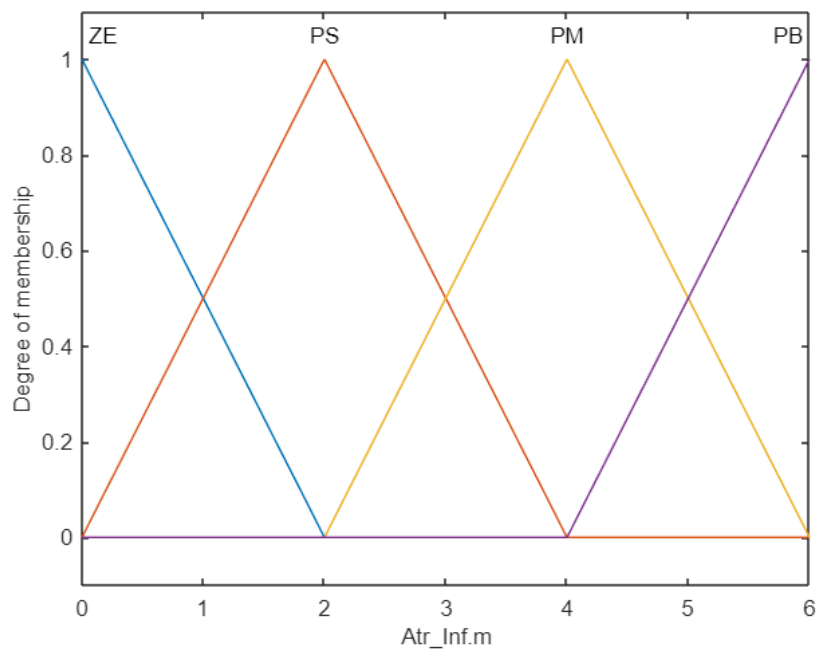
```
plotmf(T1_42_FLC_Anest(2,6), 'input', 2);
```



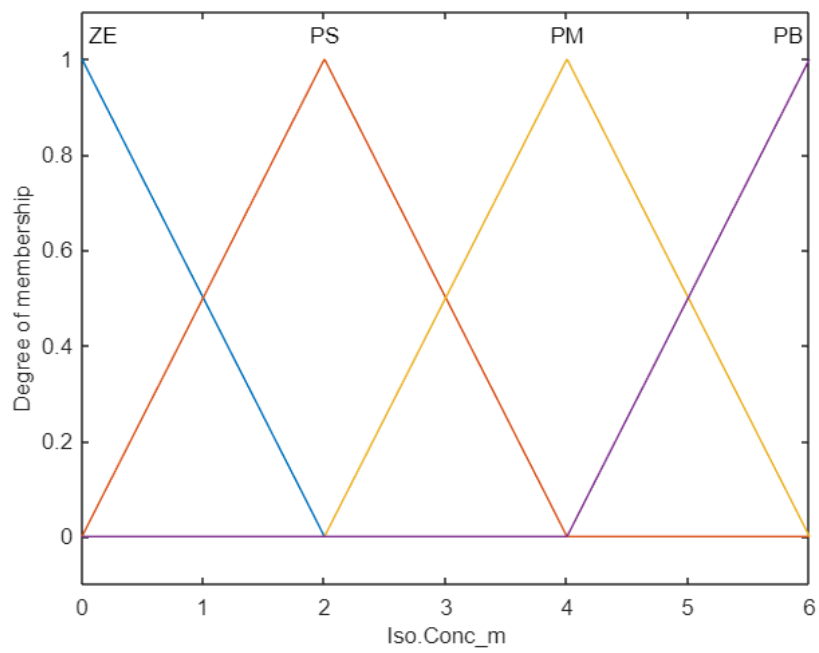
```
plotmf(T1_42_FLC_Anest(1,1), 'output', 1);
```



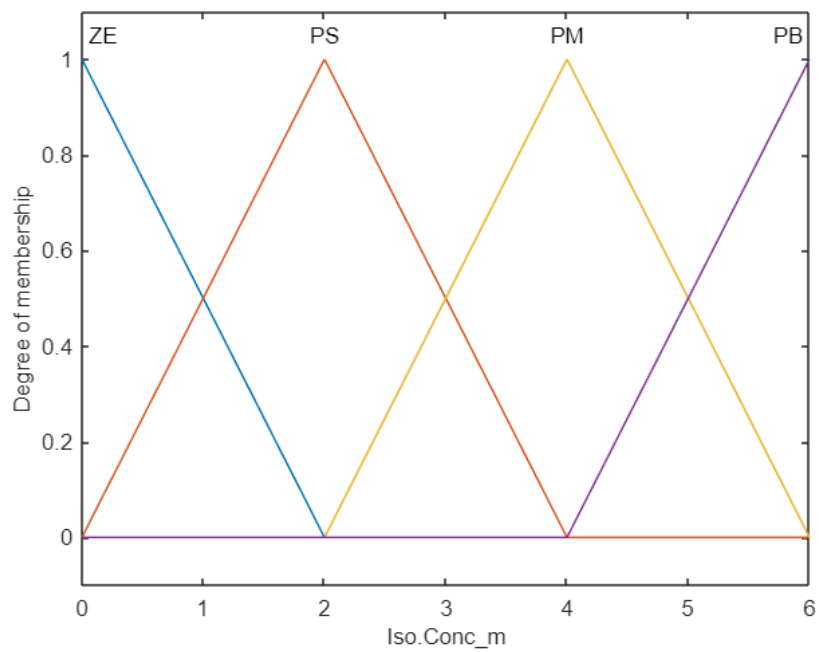
```
plotmf(T1_42_FLC_Anest(1,6), 'output', 1);
```



```
plotmf(T1_42_FLC_Anest(2,1), 'output', 1);
```



```
plotmf(T1_42_FLC_Anest(2,6), 'output',1);
```



```
disp(T1_42_FLC_Anest(1,1));
```

mamfis with properties:

```
Name: "Controlador 11 T1FLC"
AndMethod: "min"
OrMethod: "max"
```

```
ImplicationMethod: "min"
AggregationMethod: "max"
DefuzzificationMethod: "centroid"
    Inputs: [1x2 fisvar]
    Outputs: [1x1 fisvar]
    Rules: [1x25 fisrule]
DisableStructuralChecks: 0
```

See 'getTunableSettings' method for parameter optimization.

```
disp(T1_42_FLC_Anest(1,6));
```

mamfis with properties:

```
    Name: "Controlador 61 T1FLC"
    AndMethod: "min"
    OrMethod: "max"
    ImplicationMethod: "min"
    AggregationMethod: "max"
    DefuzzificationMethod: "centroid"
        Inputs: [1x2 fisvar]
        Outputs: [1x1 fisvar]
        Rules: [1x25 fisrule]
DisableStructuralChecks: 0
```

See 'getTunableSettings' method for parameter optimization.

```
disp(T1_42_FLC_Anest(2,1));
```

mamfis with properties:

```
    Name: "Controlador 12 T1FLC"
    AndMethod: "min"
    OrMethod: "max"
    ImplicationMethod: "min"
    AggregationMethod: "max"
    DefuzzificationMethod: "centroid"
        Inputs: [1x2 fisvar]
        Outputs: [1x1 fisvar]
        Rules: [1x25 fisrule]
DisableStructuralChecks: 0
```

See 'getTunableSettings' method for parameter optimization.

```
disp(T1_42_FLC_Anest(2,6));
```

mamfis with properties:

```
        Name: "Controlador 62 T1FLC"  
        AndMethod: "min"  
        OrMethod: "max"  
        ImplicationMethod: "min"  
        AggregationMethod: "max"  
        DefuzzificationMethod: "centroid"  
        Inputs: [1x2 fisvar]  
        Outputs: [1x1 fisvar]  
        Rules: [1x25 fisrule]  
DisableStructuralChecks: 0
```

See 'getTunableSettings' method for parameter optimization.

```
save T1_42_FLC.mat T1_42_FLC_Anest;
```

Função para gerar o controlador Fuzzy Tipo 2 Intervalar (IT2_22_FLC)

Autor: Giovanni Oliveira de Sousa

Ano: 2021

2 ENTRADAS: RM_E, PS_E

2 SAÍDAS: ATRAC_INF, ISO_CONC

```
clc
clear
close all
% O controlador é baseado em SHIEH et al. (2009)
% https://doi.org/10.1007/978-3-540-89968-6_14
% Os conjuntos Fuzzy Tipo 2 Intervalar são baseados em DOCTOR et al.
(2016)
% https://doi.org/10.1016/j.asoc.2015.10.014
% O controlador possui 2 entradas (inputs) e 1 saída (outputs);
% O controlador será decomposto em 2 controladores simples;
% Um controlador simples será dedicado para uma saída;
% Logo, cada controlador simples possui 2 entradas e 1 saída;
% Cada entrada possui 7 funções de pertinência (MF) tipo 2
intervalar;
% Cada saída possui 4 funções de pertinência (MF) tipo 2 intervalar;
% Os controladores será declarado sem regras;
% O controlador utilizará o método de redução de ordem EIASC.
IT2_22_FLC_Anest(1:2) = mamfistype2; % Vetor nulo de controladores fuzzy
tipo-2
Nome_ctr = ["Controlador 1 IT2FLC","Controlador 2 IT2FLC"]; % Nome de
cada controlador
for i = 1:2
    IT2_22_FLC_Anest(i) =
mamfistype2("Name",Nome_ctr(i),'NumInputs',2,'NumInputMFs',7,...
'NumOutputs',1,'NumOutputMFs',4,'AddRules','none','TypeReductionMethod',"
eiasc");
end
```

1. DEFINIR PARÂMETROS FIXOS:

A. NOME (STRING) DAS ENTRADAS (INPUTS) E SAÍDAS (INPUTS):

Entradas: Relaxamento Muscular erro (**RM_e**); Relaxamento Muscular integral erro (**RM_int.e**).

Saídas: Pressão Sanguínea erro (**PS_e**); Pressão Sanguínea integral erro (**PS_int.e**).

B. NOME (STRING) DAS FUNÇÕES DE PERTINÊNCIA (MFS) DAS ENTRADAS E SAÍDAS:

Entradas: Negativo Grande (**NB**); Negativo Médio (**NM**); Negativo Pequeno (**NS**); Zero (**ZE**); Positivo Pequeno (**PS**); Positivo Médio (**PM**); Positivo Grande (**PB**).

Saídas: Zero (**ZE**); Positivo Pequeno (**PS**); Positivo Médio (**PM**); Positivo Grande (**PB**).

C. VALORES DOS CONJUNTOS FUZZY (FUNÇÕES DE PERTINÊNCIA - MF) NOMINAIS TIPO 1:

Os conjuntos nominais são MFs triangulares com largura de 4 e sobreposição de 25%:

NS - [-4 -2 0];

ZE - [-2 0 2];

PS - [0 2 4];

...

D. DESLOCAMENTO DE CADA MF DE CADA ENTRADA E SAÍDA (FORNECIDO PELA EQUAÇÃO DA REFERÊNCIA).

O que define um conjunto Fuzzy Tipo-2 Intervalar é sua mancha de incerteza (FOU - *Footprint of Uncertainty*) e isso pode ser definido como um deslocamento bidirecional d em relação a toda a função de pertinência. Exemplo: Para um deslocamento de 0,3 em um certo conjunto fuzzy tipo-2 intervalar que tem função de pertinência nominal com valor 0,5 para um certo x. O valor real para x será entre 0,5 - 0,3 até 0,5 + 0,3, ou seja, de 0,2 a 0,8.

Doctor et al. (2016) demonstra um método heurístico para gerar a mancha de incerteza no controle fuzzy tipo-2 de anestesia utilizando dados de pacientes anestesiados de procedimentos cirúrgicos:

Primeiro é preciso calcular o coeficiente de variação dado pela fórmula:

$$cv_{\partial}^{(t)} = \frac{stdv_{\partial}^{(t)}}{avg_{\partial}^{(t)}} \times 100$$

A variável ∂ significa a saída a ser controlada (RM - Relaxamento muscular ou PS - Pressão Sanguínea). stdv e avg significam o desvio padrão e média da variável ∂ , respectivamente, e são obtidos por meio de dados experimentais. A variável t significa o número do paciente e tem uma faixa $0 \leq t \leq T$, onde T é o número de pacientes obtidos nos dados experimentais. A referência citada anteriormente fornece uma tabela com dados de 15 pacientes e utiliza a média de cada valor para calcular os seguintes valores:

$$cv_{rm}^{(media)} = \frac{10.95}{89.69} \times 100 = 12.21$$

$$cv_{ps}^{(media)} = \frac{12.40}{90.88} \times 100 = 13.64$$

O próximo passo é calcular valores de acordo com a referência de controle da saída (80% de Relaxamento Muscular e 110mmHG de Pressão Sanguínea). O coeficientes de variação em escala de cada um será:

$$cv\Delta_{rm}^{(media)} = \frac{cv_{rm}^{(media)}}{100} \times 80 = 12.21 \times 0.8 = 9.77$$

$$cv\Delta_{ps}^{(media)} = \frac{cv_{ps}^{(media)}}{100} \times 110 = 13.64 \times 1.1 = 15$$

Por último o cálculo da mancha da incerteza é a divisão do coeficiente de variação pela faixa de valores possíveis da entrada em escala pr_{∂} que será para as duas saídas [-7,7], ou seja, $pr_{rm} = pr_{ps} = \pm 7$. Para a construção do conjunto fuzzy tipo-2 intervalar, é necessário estender o valor nominal com a mancha da incerteza, isso é feito somando e subtraindo com a metade valor da mancha de incerteza simetricamente com o valor nominal, portanto é calculado o valor de incerteza simétrico para cada saída:

$$\pm uc_{rm} = \frac{(cv\Delta_{rm}^{(media)})/pr_{mr}}{2} = \frac{9.77/7}{2} = \pm 0.7$$

$$\pm uc_{ps} = \frac{(cv\Delta_{ps}^{(media)})/pr_{ps}}{2} = \frac{15/7}{2} = \pm 1$$

Esses valores (Desl_MFInputs) serão utilizados para compor os conjuntos Fuzzy tipo-2 intervalares de entrada nos controladores. A referência citada anteriormente não utiliza nenhum método para os conjuntos de saída, portanto, será utilizado um valor a partir da tentativa e erro.

```
% Entradas
Nome_inputs = ["RM.e", "PS.e"];
Desl_MFInputs = [ 0.7 , 1 ];

NMFs_inputs = [ "NB" , "NM" , "NS" , "ZE" , "PS" ,
"PM" , "PB" ];
ValNom_MFInputs = [ -8 -6 -4 ; -6 -4 -2 ; -4 -2 0 ; -2 0 2 ; 0 2 4 ; 2 4
6 ; 4 6 8 ];

% Saídas
Nome_outputs = ["Atr.inf", "Iso.conc"];
Desl_MFOutputs = [ 0.65 , 0.1 ];

NMFs_outputs = [ "ZE" , "PS" , "PM" , "PB" ];
ValNom_MFOutputs = [ -2 0 2 ; 0 2 4 ; 2 4 6 ; 4 6 8 ];
```

2. CALCULAR OS VALORES DE MFS TIPO 2:

A partir do valor nominal de MFs de tipo 1, serão calculados os valores (**ValCalc_MFI/O**) de tipo 2, partindo do princípio de que a MF nominal seria deslocada pelas variáveis (**Desl_MFI/O**) tanto para esquerda e direita, formando o FOU.

Os formatos resultantes são trapezoidais com quatro valores.

```
% Entradas: 7 MFs e 4 valores para o formato trapezoidal
% Os valores de RM e PM terão deslocamentos diferentes
% O vetor contendo os valores será de 14 x 4
ValCalc_MFInputs = zeros(14,4);
for i = 1:2
    for j = 1:7
        d = 7*(i-1);
        val1 = ValNom_MFInputs(j,1) - Desl_MFInputs(i);
        val2 = ValNom_MFInputs(j,2) - Desl_MFInputs(i);
        val3 = ValNom_MFInputs(j,2) + Desl_MFInputs(i);
        val4 = ValNom_MFInputs(j,3) + Desl_MFInputs(i);
        if j == 1
            ValCalc_MFInputs(j+d,:) = [val2, val2, val3, val4];
        elseif j == 7
            ValCalc_MFInputs(j+d,:) = [val1, val2, val3, val3];
        else
            ValCalc_MFInputs(j+d,:) = [val1, val2, val3, val4];
        end
    end
end

% Saídas: 4 MFs e 4 valores para o formato trapezoidal
% Os valores de Atracúrio e Isoflurano terão deslocamentos diferentes
% O vetor contendo os valores será de 8 x 4
ValCalc_MFOutputs = zeros(8,4);
for i = 1:2
    for j = 1:4
        d = 4*(i-1);
        val1 = ValNom_MFOutputs(j,1) - Desl_MFOutputs(i);
        val2 = ValNom_MFOutputs(j,2) - Desl_MFOutputs(i);
        val3 = ValNom_MFOutputs(j,2) + Desl_MFOutputs(i);
        val4 = ValNom_MFOutputs(j,3) + Desl_MFOutputs(i);
        if j == 1
            ValCalc_MFOutputs(j+d,:) = [val2, val2, val3, val4];
        elseif j == 4
            ValCalc_MFOutputs(j+d,:) = [val1, val2, val3, val3];
        else
            ValCalc_MFOutputs(j+d,:) = [val1, val2, val3, val4];
        end
    end
end
```

3. DEFINIR OS PARÂMETROS DE CADA MF DA I/O

A. DEFINIR OS NOMES (STRINGS) DE CADA ENTRADA E SAÍDA;

B. DEFINIR A FAIXA DE VALORES (UNIVERSO DE DISCURSO) DE CADA ENTRADA E SAÍDA;

C. DEFINIR CADA FUNÇÃO DE PERTINÊNCIA (MF) COMO TIPO TRIANGULAR (TRIMF);

Aqui as funções de pertinência (MFs) são definidas como tipo trapezoidal (*trapmf*) com seus parâmetros fixos. A faixa de valores está entre -7 e 7, porém, só serve para as entradas que admitem valores negativos. As saídas só podem aceitar valores positivos, logo sua faixa está entre 0 e 7.

```
% Entradas: 2 entradas com 7 MFs
for i = 1:2 % Controladores para todas as saídas
    for j = 1:2 % Entradas para cada controlador
        IT2_22_FLC_Anest(i).Inputs(j).Name = Nome_inputs(j);
        IT2_22_FLC_Anest(i).Inputs(j).Range = [-7 7];
        for k = 1:7 % Funções de Pertinência por entrada em cada
            controlador
                d = 7*(j-1);
                IT2_22_FLC_Anest(i).Inputs(j).MembershipFunctions(k).Name =
                    NMFs_inputs(k);
                IT2_22_FLC_Anest(i).Inputs(j).MembershipFunctions(k).Type =
                    "trapmf";

                IT2_22_FLC_Anest(i).Inputs(j).MembershipFunctions(k).UpperParameters =
                    ...
                        ValCalc_MFInputs(k+d,:);

                IT2_22_FLC_Anest(i).Inputs(j).MembershipFunctions(k).LowerScale = 1;

                IT2_22_FLC_Anest(i).Inputs(j).MembershipFunctions(k).LowerLag =
                    Desl_MFInputs(j);
            end
        end
    end

% Saídas: 1 saída com 4 MFs
for i = 1:2 % Controladores para todas as saídas
    IT2_22_FLC_Anest(i).Outputs(1).Name = Nome_outputs(i);
    IT2_22_FLC_Anest(i).Outputs(1).Range = [0 7];
    for j = 1:4 % Funções de Pertinência por saída em cada controlador
        d = 4*(i-1);
        IT2_22_FLC_Anest(i).Outputs(1).MembershipFunctions(j).Name =
            NMFs_outputs(j);
        IT2_22_FLC_Anest(i).Outputs(1).MembershipFunctions(j).Type =
            "trapmf";

        IT2_22_FLC_Anest(i).Outputs(1).MembershipFunctions(j).UpperParameters =
            ...
```

```

        ValCalc_MFOutputs(j+d,:);
        IT2_22_FLC_Anest(i).Outputs(1).MembershipFunctions(j).LowerScale
= 1;
        IT2_22_FLC_Anest(i).Outputs(1).MembershipFunctions(j).LowerLag =
Desl_MFOutputs(i);
    end
end

```

4. DEFINIR A BASE DE REGRAS

As regras já foram definidas na referência de acordo com as imagens abaixo. Para definir no programa é necessário criar uma matriz com o tamanho correto de combinações possíveis. Cada regra terá o formato de:

SE Entrada (1,2) == (NB - 1 ... PB - 7) **E** Entrada (1,2) == (NB - 1 ... PB - 7) **ENTÃO** Saída (1,2) == (ZE - 1 ... PB - 4)

Cada regra pode ser descrita somente com os índices de entrada(s), função de pertinência(s) e saída(s).

Cada controlador terá uma base de regras distinta, ou seja, para cada saída há uma base de regras.

Atra_Inf	NB	NM	NS	ZE	PS	PM	PB
NB	PB		PS		ZE		ZE
NM		PM		ZE		ZE	
NS	PB		PS		ZE		ZE
ZE		PM		ZE		ZE	
PS	PM		ZE		ZE		ZE
PM		PM		ZE		ZE	
PB	PM		ZE		ZE		ZE

Fonte: SHIEH et al. (2009)

Iso_Conc	NB	NM	NS	ZE	PS	PM	PB
NB	ZE		ZE		ZE		ZE
NM		ZE		ZE		ZE	
NS	ZE		ZE		ZE		ZE
ZE		ZE		ZE		ZE	
PS	PS		PS		PS		ZE
PM		PM		PS		ZE	
PB	PB		PM		PS		PS

Fonte: SHIEH et al. (2009)

```

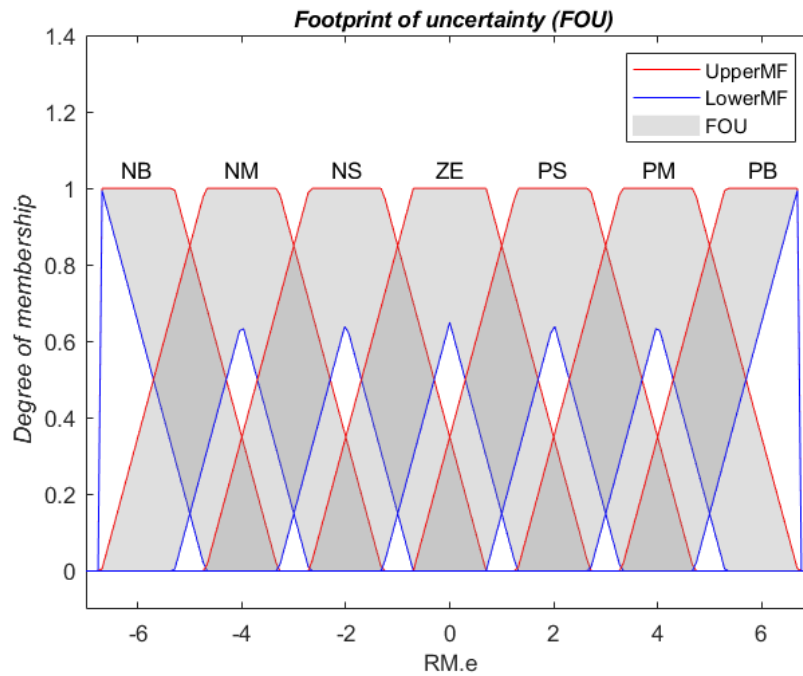
Regras_Fuzzy_IT2_22 = zeros(25,5*2);
Regras_Fuzzy_IT2_22(:,4:5) = 1;
Regras_Fuzzy_IT2_22(:,9:10) = 1;
O1 = [4 2 1 1 , 3 1 1 , 4 2 1 1 , 3 1 1 , 3 1 1 1 , 3 1 1 , 3 1 1 1];
O2 = [1 1 1 1 , 1 1 1 , 1 1 1 1 , 1 1 1 , 2 2 2 1 , 3 2 1 , 4 3 2 2];
O = [O1;O2];
cont1 = 1;
cont2 = 1;
for i = 1:25
    for j = 1:2
        d = 5*(j-1);
        Regras_Fuzzy_IT2_22(i,1+d) = cont1;
        Regras_Fuzzy_IT2_22(i,2+d) = cont2;
        Regras_Fuzzy_IT2_22(i,3+d) = O(j,i);
    end
    if cont1+2 <= 7
        cont1 = cont1 + 2;
    else
        if cont1+2 == 8
            cont1 = 1;
        else
            cont1 = 2;
        end
        cont2 = cont2 + 1;
    end
end
for i = 1:2
    d = 5*(i-1);
    IT2_22_FLC_Anest(i) =
addRule(IT2_22_FLC_Anest(i),Regras_Fuzzy_IT2_22(1:25,1+d:5+d));
end
Fuzzifi_MFs(1:2) = figure;

```

```

Fuzzifi_MFs(1) = figure('Name','Fuzzificação RM.e','NumberTitle','off');
plotmf(IT2_22_FLC_Anest(1),'input',1);
title("\it Footprint of uncertainty (FOU)");
ylabel("\it Degree of membership");
exportgraphics(Fuzzifi_MFs(1),'Fuzzificacao_RM_e.tif','Resolution',600);

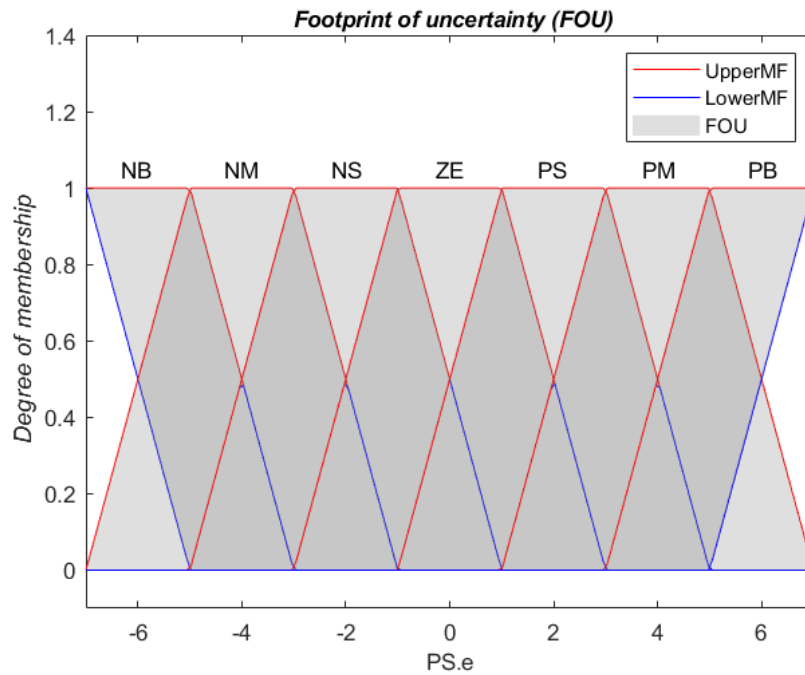
```



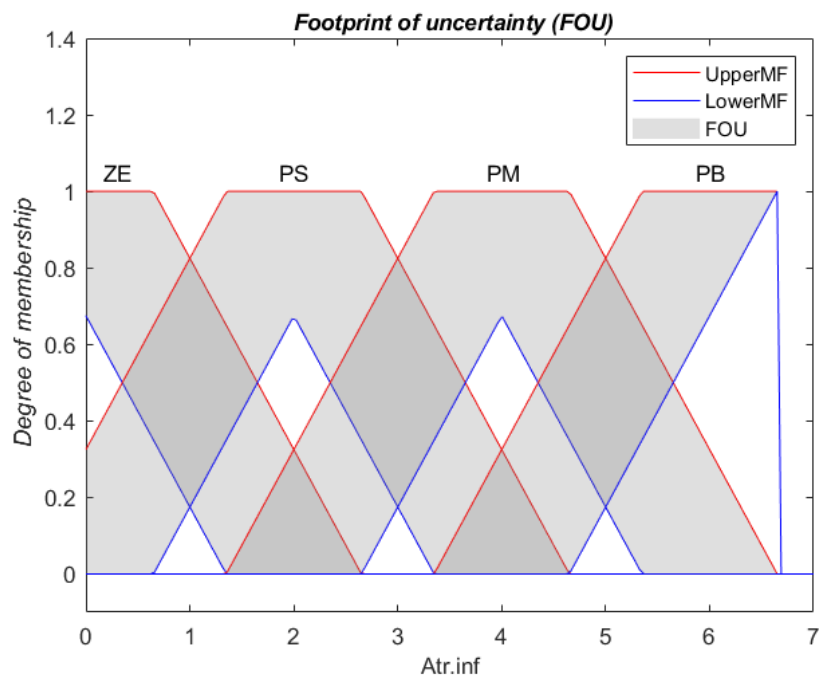
```

Fuzzifi_MFs(2) = figure('Name','Fuzzificação PS.e','NumberTitle','off');
plotmf(IT2_22_FLC_Anest(1),'input',2);
title("\it Footprint of uncertainty (FOU)");
ylabel("\it Degree of membership");
exportgraphics(Fuzzifi_MFs(2),'Fuzzificacao_PS_e.tif','Resolution',600);

```



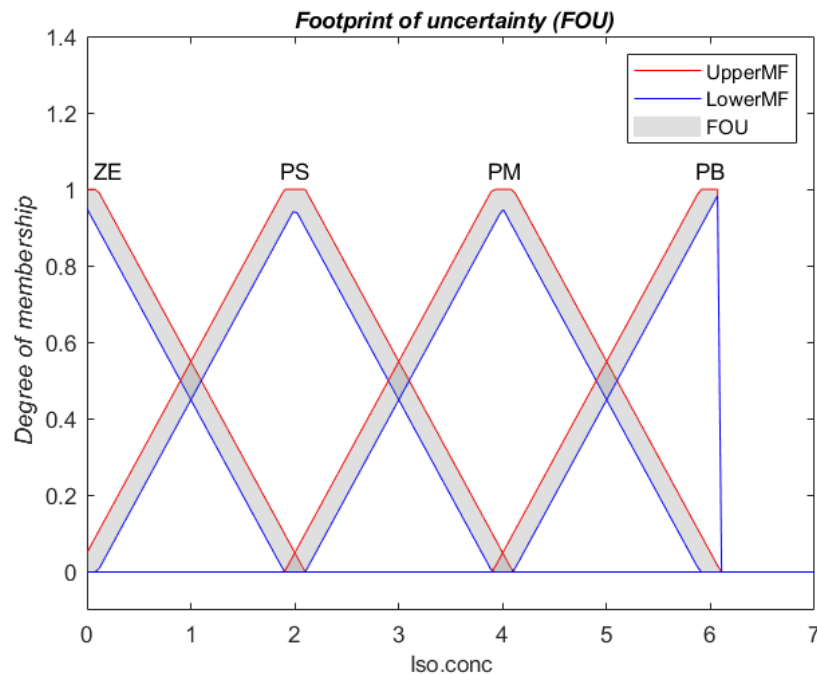
```
Defuzzi_MFs(1:2) = figure;
Defuzzi_MFs(1) = figure('Name','Defuzzificação
Atr.inf','NumberTitle','off');
plotmf(IT2_22_FLC_Anest(1),'output',1);
title("\it Footprint of uncertainty (FOU)");
ylabel("\it Degree of membership");
exportgraphics(Defuzzi_MFs(1),'Defuzzificacao_Atr_inf.tif','Resolution',
600);
```



```

Defuzzi_MFs(2) = figure('Name','Defuzzificação
Iso.conc','NumberTitle','off');
plotmf(IT2_22_FLC_Anest(2),'output',1);
title("\it Footprint of uncertainty (FOU)");
ylabel("\it Degree of membership");
exportgraphics(Defuzzi_MFs(2),'Defuzzificacao_Iso_conc.tif','Resolution'
,600);

```



```

disp(IT2_22_FLC_Anest(1));

```

mamfistype2 with properties:

```

        Name: "Controlador 1 IT2FLC"
        AndMethod: "min"
        OrMethod: "max"
        ImplicationMethod: "min"
        AggregationMethod: "max"
        DefuzzificationMethod: "centroid"
        Inputs: [1x2 fisvar]
        Outputs: [1x1 fisvar]
        Rules: [1x25 fisrule]
        DisableStructuralChecks: 0
        TypeReductionMethod: "eiasc"

```

See 'getTunableSettings' method for parameter optimization.

```

disp(IT2_22_FLC_Anest(2));

```

mamfistype2 with properties:

```
        Name: "Controlador 2 IT2FLC"  
        AndMethod: "min"  
        OrMethod: "max"  
        ImplicationMethod: "min"  
        AggregationMethod: "max"  
        DefuzzificationMethod: "centroid"  
        Inputs: [1x2 fisvar]  
        Outputs: [1x1 fisvar]  
        Rules: [1x25 fisrule]  
DisableStructuralChecks: 0  
TypeReductionMethod: "eiasc"
```

See 'getTunableSettings' method for parameter optimization.

```
save IT2_22_FLC.mat IT2_22_FLC_Anest;
```

Função para gerar o controlador Fuzzy Tipo 2 Intervalar (IT2_42_FLC)

Autor: Giovanni Oliveira de Sousa

Ano: 2021

4 ENTRADAS: RM_E, RM_INT.E, PS_E, PS_INT.E

2 SAÍDAS: ATRAC_INF, ISO_CONC

```
clc
clear
close all
% O controlador é baseado em SHIEH et al. (2009)
% https://doi.org/10.1007/978-3-540-89968-6_14
% Os conjuntos Fuzzy Tipo 2 Intervalar são baseados em DOCTOR et al.
(2016)
% https://doi.org/10.1016/j.asoc.2015.10.014
% O controlador possui 2 entradas (inputs) e 1 saída (outputs);
% O controlador será decomposto em 2 controladores simples;
% Um controlador simples será dedicado para uma saída;
% Logo, cada controlador simples possui 2 entradas e 1 saída;
% Cada entrada possui 7 funções de pertinência (MF) tipo 2
intervalar;
% Cada saída possui 4 funções de pertinência (MF) tipo 2 intervalar;
% Os controladores será declarado sem regras;
% O controlador utilizará o método de redução de ordem EIASC.
IT2_42_FLC_Anest(1:2,1:6) = mamfistype2; % Vetor nulo de controladores
fuzzy tipo-2
% Nome de cada controlador
Nome_ctr = ["Controlador 11 IT2FLC","Controlador 21 IT2FLC","Controlador
31 IT2FLC",...
            "Controlador 41 IT2FLC","Controlador 51 IT2FLC","Controlador
61 IT2FLC";...
            "Controlador 12 IT2FLC","Controlador 22 IT2FLC","Controlador
32 IT2FLC",...
            "Controlador 42 IT2FLC","Controlador 52 IT2FLC","Controlador
62 IT2FLC"];
for i = 1:6
    for j = 1:2
        IT2_42_FLC_Anest(j,i) =
mamfistype2("Name",Nome_ctr(j,i),'NumInputs',2,'NumOutputs'...
,1,'NumInputMFs',7,'NumOutputMFs',4,'AddRules','none','TypeReductionMetho
d',"eiasc");
```

end
end

1. DEFINIR PARÂMETROS FIXOS:

A. NOME (STRING) DAS ENTRADAS (INPUTS) E SAÍDAS (INPUTS):

Entradas: Relaxamento Muscular erro (**RM_e**); Relaxamento Muscular integral erro (**RM_int.e**).

Saídas: Pressão Sanguínea erro (**PS_e**); Pressão Sanguínea integral erro (**PS_int.e**).

B. NOME (STRING) DAS FUNÇÕES DE PERTINÊNCIA (MFS) DAS ENTRADAS E SAÍDAS:

Entradas: Negativo Grande (**NB**); Negativo Médio (**NM**); Negativo Pequeno (**NS**); Zero (**ZE**); Positivo Pequeno (**PS**); Positivo Médio (**PM**); Positivo Grande (**PB**).

Saídas: Zero (**ZE**); Positivo Pequeno (**PS**); Positivo Médio (**PM**); Positivo Grande (**PB**).

C. VALORES DOS CONJUNTOS FUZZY (FUNÇÕES DE PERTINÊNCIA - MF) NOMINAIS TIPO 1:

Os conjuntos nominais são MFs triangulares com largura de 4 e sobreposição de 25%:

NS - [-4 -2 0];

ZE - [-2 0 2];

PS - [0 2 4];

...

D. DESLOCAMENTO DE CADA MF DE CADA ENTRADA E SAÍDA (FORNECIDO PELA EQUAÇÃO DA REFERÊNCIA).

O que define um conjunto Fuzzy Tipo-2 Intervalar é sua mancha de incerteza (FOU - *Footprint of Uncertainty*) e isso pode ser definido como um deslocamento bidirecional d em relação a toda a função de pertinência. Exemplo: Para um deslocamento de 0,3 em um certo conjunto fuzzy tipo-2 intervalar que tem função de pertinência nominal com valor 0,5 para um certo x. O valor real para x será entre 0,5 - 0,3 até 0,5 + 0,3, ou seja, de 0,2 a 0,8.

Doctor et al. (2016) demonstra um método heurístico para gerar a mancha de incerteza no controle fuzzy tipo-2 de anestesia utilizando dados de pacientes anestesiados de procedimentos cirúrgicos:

Primeiro é preciso calcular o coeficiente de variação dado pela fórmula:

$$cv_{\partial}^{(t)} = \frac{stdv_{\partial}^{(t)}}{avg_{\partial}^{(t)}} \times 100$$

A variável ∂ significa a saída a ser controlada (RM - Relaxamento muscular ou PS - Pressão Sanguínea). stdv e avg significam o desvio padrão e média da variável ∂ , respectivamente, e são obtidos por meio de dados experimentais. A variável t significa o número do paciente e tem uma faixa $0 \leq t \leq T$, onde T é o número de pacientes obtidos nos dados experimentais. A referência citada anteriormente fornece uma tabela com dados de 15 pacientes e utiliza a média de cada valor para calcular os seguintes valores:

$$cv_{rm}^{(media)} = \frac{10.95}{89.69} \times 100 = 12.21$$

$$cv_{ps}^{(media)} = \frac{12.40}{90.88} \times 100 = 13.64$$

O próximo passo é calcular valores de acordo com a referência de controle da saída (80% de Relaxamento Muscular e 110mmHG de Pressão Sanguínea). O coeficientes de variação em escala de cada um será:

$$cv\Delta_{rm}^{(media)} = \frac{cv_{rm}^{(media)}}{100} \times 80 = 12.21 \times 0.8 = 9.77$$

$$cv\Delta_{ps}^{(media)} = \frac{cv_{ps}^{(media)}}{100} \times 110 = 13.64 \times 1.1 = 15$$

Por último o cálculo da mancha da incerteza é a divisão do coeficiente de variação pela faixa de valores possíveis da entrada em escala pr_{∂} que será para as duas saídas $[-7,7]$, ou seja, $pr_{rm} = pr_{ps} = \pm 7$. Para a construção do conjunto fuzzy tipo-2 intervalar, é necessário estender o valor nominal com a mancha da incerteza, isso é feito somando e subtraindo com a metade valor da mancha de incerteza simetricamente com o valor nominal, portanto é calculado o valor de incerteza simétrico para cada saída:

$$\pm uc_{rm} = \frac{(cv\Delta_{rm}^{(media)})/pr_{rm}}{2} = \frac{9.77/7}{2} = \pm 0.7$$

$$\pm uc_{ps} = \frac{(cv\Delta_{ps}^{(media)})/pr_{ps}}{2} = \frac{15/7}{2} = \pm 1$$

Esses valores (Desl_MFInputs) serão utilizados para compor os conjuntos Fuzzy tipo-2 intervalares de entrada nos controladores. A referência citada anteriormente não utiliza nenhum método para os conjuntos de saída (Desl_MFOutputs), portanto, será utilizado um valor a partir da tentativa e erro.

As entradas integrais do erro (RM_int.e e PS_int.e) utilizarão o mesmo valor de seus respectivos erros.

```
% Entradas
Nome_inputs = ["RM_e", "RM_{int.e}", "PS_e", "PS_{int.e}"];
Desl_MFInputs = [ 0.7 , 0.7 , 1 , 1 ];

Ordem_inputs = [1 1 1 , 2 2 3 ; 2 3 4 , 3 4 4]; % Ordem das entradas
```

```

NMFs_inputs = [ "NB" , "NM" , "NS" , "ZE" , "PS" ,
"PM" , "PB" ];
ValNom_MFInputs = [ -8 -6 -4 ; -6 -4 -2 ; -4 -2 0 ; -2 0 2 ; 0 2 4 ; 2 4
6 ; 4 6 8 ];

% Saídas
Nome_outputs = ["Atr_{Inf}", "Iso_{Conc}"];
Desl_MFOutputs = [ 0.1 , 0.1 ];

NMFs_outputs = [ "ZE" , "PS" , "PM" , "PB" ];
ValNom_MFOutputs = [ -2 0 2 ; 0 2 4 ; 2 4 6 ; 4 6 8 ];

```

2. CALCULAR OS VALORES DE MFS TIPO 2:

A partir do valor nominal de MFs de tipo 1, serão calculados os valores (**ValCalc_MFI/O**) de tipo 2, partindo do princípio de que a MF nominal seria deslocada pelas variáveis (**Desl_MFI/O**) tanto para esquerda e direita, formando o FOU.

Os formatos resultantes são trapezoidais com quatro valores.

```

% Entradas: 7 MFs e 4 valores para o formato trapezoidal
% Os valores de RM e PS terão deslocamentos diferentes
% Os valores de RM_int e PS serão os mesmos dos anteriores
% O vetor contendo os valores será de 28 x 4
ValCalc_MFInputs = zeros(28,4);
for i = 1:4
    for j = 1:7
        d = 7*(i-1);
        val1 = ValNom_MFInputs(j,1) - Desl_MFInputs(i);
        val2 = ValNom_MFInputs(j,2) - Desl_MFInputs(i);
        val3 = ValNom_MFInputs(j,2) + Desl_MFInputs(i);
        val4 = ValNom_MFInputs(j,3) + Desl_MFInputs(i);
        if j == 1
            ValCalc_MFInputs(j+d,:) = [val2, val2, val3, val4];
        elseif j == 7
            ValCalc_MFInputs(j+d,:) = [val1, val2, val3, val3];
        else
            ValCalc_MFInputs(j+d,:) = [val1, val2, val3, val4];
        end
    end
end

% Saídas: 4 MFs e 4 valores para o formato trapezoidal
% Os valores de Atracúrio e Isoflurano terão deslocamentos diferentes
% O vetor contendo os valores será de 8 x 4
ValCalc_MFOutputs = zeros(8,4);
for i = 1:2
    for j = 1:4
        d = 4*(i-1);
        val1 = ValNom_MFOutputs(j,1) - Desl_MFOutputs(i);

```

```

        val2 = ValNom_MFOutputs(j,2) - Desl_MFOutputs(i);
        val3 = ValNom_MFOutputs(j,2) + Desl_MFOutputs(i);
        val4 = ValNom_MFOutputs(j,3) + Desl_MFOutputs(i);
        if j == 1
            ValCalc_MFOutputs(j+d,:) = [val2, val2, val3, val4];
        elseif j == 4
            ValCalc_MFOutputs(j+d,:) = [val1, val2, val3, val3];
        else
            ValCalc_MFOutputs(j+d,:) = [val1, val2, val3, val4];
        end
    end
end

```

3. DEFINIR OS PARÂMETROS DE CADA MF DA I/O

- A. DEFINIR OS NOMES (STRINGS) DE CADA ENTRADA E SAÍDA;
- B. DEFINIR A FAIXA DE VALORES (UNIVERSO DE DISCURSO) DE CADA ENTRADA E SAÍDA;
- C. DEFINIR CADA FUNÇÃO DE PERTINÊNCIA (MF) COMO TIPO TRIANGULAR (TRIMF);

Aqui as funções de pertinência (MFs) são definidas como tipo triangular (*trapmf*) com seus parâmetros fixos. A faixa de valores está entre -7 e 7, porém, só serve para as entradas que admitem valores negativos. As saídas só podem aceitar valores positivos, logo sua faixa está entre 0 e 7.

```

% Entradas: 2 entradas com 7 MFs
for i = 1:6 % Controladores por saída
    for j = 1:2 % Saídas totais
        for k = 1:2 % Entradas em cada controlador
            IT2_42_FLC_Anest(j,i).Inputs(k).Name =
Nome_inputs(Ordem_inputs(k,i));
            IT2_42_FLC_Anest(j,i).Inputs(k).Range = [-7 7];
            for p = 1:7 % Funções de Pertinência por entrada em cada
controlador
                d = 7*(Ordem_inputs(k,i)-1);

IT2_42_FLC_Anest(j,i).Inputs(k).MembershipFunctions(p).Name =
NMFs_inputs(p);

IT2_42_FLC_Anest(j,i).Inputs(k).MembershipFunctions(p).Type = "trapmf";

IT2_42_FLC_Anest(j,i).Inputs(k).MembershipFunctions(p).UpperParameters =
...
                ValCalc_MFInputs(p+d,:);

IT2_42_FLC_Anest(j,i).Inputs(k).MembershipFunctions(p).LowerScale = 1;

```

```

IT2_42_FLC_Anest(j,i).Inputs(k).MembershipFunctions(p).LowerLag = ...
    Desl_MFInputs(Ordem_inputs(k,i));
        end
    end
end
end

% Saídas: 1 saída com 4 MFs
for i = 1:6 % Controladores por saída
    for j = 1:2 % Saídas totais (uma saída por controlador)
        IT2_42_FLC_Anest(j,i).Outputs(1).Name = Nome_outputs(j);
        IT2_42_FLC_Anest(j,i).Outputs(1).Range = [0 7];
        for k = 1:4 % Funções de Pertinência por saída em cada
controlador
            d = 4*(j-1);
            IT2_42_FLC_Anest(j,i).Outputs(1).MembershipFunctions(k).Name
= NMFs_outputs(k);
            IT2_42_FLC_Anest(j,i).Outputs(1).MembershipFunctions(k).Type
= "trapmf";

IT2_42_FLC_Anest(j,i).Outputs(1).MembershipFunctions(k).UpperParameters =
...
                ValCalc_MFOutputs(k+d,:);

IT2_42_FLC_Anest(j,i).Outputs(1).MembershipFunctions(k).LowerScale = 1;

IT2_42_FLC_Anest(j,i).Outputs(1).MembershipFunctions(k).LowerLag = ...
                Desl_MFOutputs(j);
            end
        end
    end
end

```

4. DEFINIR A BASE DE REGRAS

As regras já foram definidas na referência de acordo com as imagens abaixo. Para definir no programa é necessário criar uma matriz com o tamanho correto de combinações possíveis. Cada regra terá o formato de:

SE Entrada (1,2,3,4) == (NB - 1 ... PB - 7) **E** Entrada (1,2,3,4) == (NB - 1 ... PB - 7) **ENTÃO**
 Saída (1,2) == (ZE - 1 ... PB - 4)

Cada regra pode ser descrita somente com os índices de entrada(s), função de pertinência(s) e saída(s).

Cada controlador terá uma base de regras distinta, ou seja, para cada saída há seis bases de regras.

Atra_Inf	NB	NM	NS	ZE	PS	PM	PB
NB	ZE/PB PB/PB		ZE/PM PS/PS PM/PS		ZE/ZE ZE/ZE		ZE/ZE ZE/ZE
NM		ZE/PM PM/PM PM/PM		ZE/ZE ZE/ZE		ZE/ZE ZE/ZE	
NS	ZE/PM PB/PB		ZE/PS PS/PS PS/PS		ZE/ZE ZE/ZE		ZE/ZE ZE/ZE
ZE		ZE/PM PM/PM PM/PM		ZE/ZE ZE/ZE		ZE/ZE ZE/ZE	
PS	ZE/PS PM/PM PB/PB		ZE/PS ZE/ZE PS/PS		ZE/ZE ZE/ZE		ZE/ZE ZE/ZE
PM		ZE/PS PM/PM PS/PM		ZE/ZE ZE/ZE		ZE/ZE ZE/ZE	
PB	ZE/PS PM/PM PB/PB		ZE/PS ZE/ZE PS/PS		ZE/ZE ZE/ZE		ZE/ZE ZE/ZE

Fonte: SHIEH et al. (2009)

Iso_Conc	NB	NM	NS	ZE	PS	PM	PB
NB	ZE/PB ZE/ZE ZE/ZE		ZE/PM ZE/ZE ZE/ZE		ZE/ZE ZE/ZE		PS/ZE ZE/ZE ZE/ZE
NM		ZE/PM ZE/ZE ZE/ZE		ZE/ZE ZE/ZE		PS/ZE ZE/ZE	
NS	ZE/PS ZE/ZE		ZE/PS ZE/ZE		PS/ZE ZE/ZE		PM/ZE ZE/ZE
ZE		ZE/ZE ZE/ZE		ZE/ZE ZE/ZE		PM/ZE ZE/ZE	
PS	ZE/ZE PS/PS PS/PM		PS/ZE PS/PS PS/PS		PS/ZE PS/PS PS/ZE		PB/ZE ZE/ZE
PM		PS/ZE PM/PM PM/PM		PM/ZE PS/PS PS/PS		PB/ZE ZE/PS	
PB	PS/ZE PB/PB PB/PB		PM/ZE PM/PM PM/PM		PB/ZE PS/PS PS/PS		PB/ZE PS/PS PS/PS

Fonte: SHIEH et al. (2009)

```

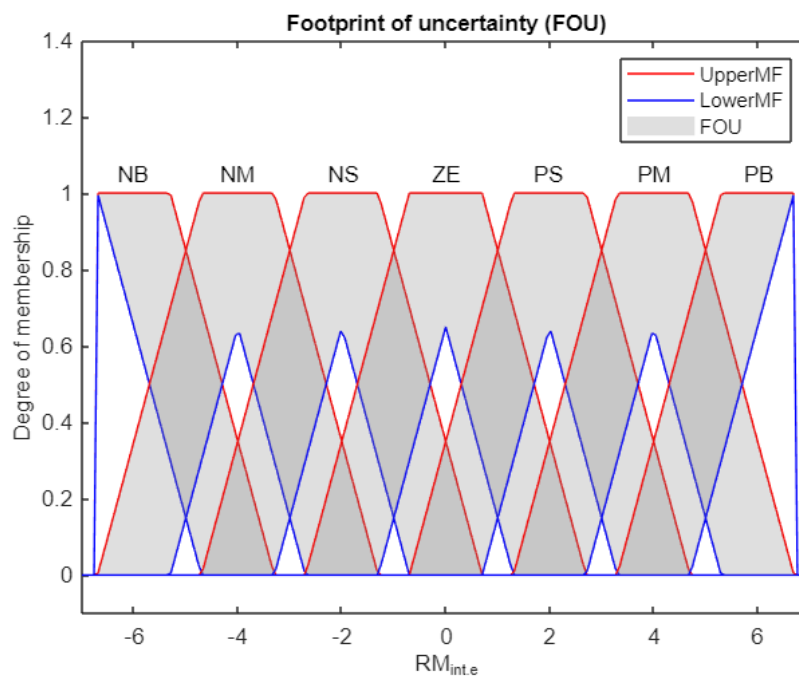
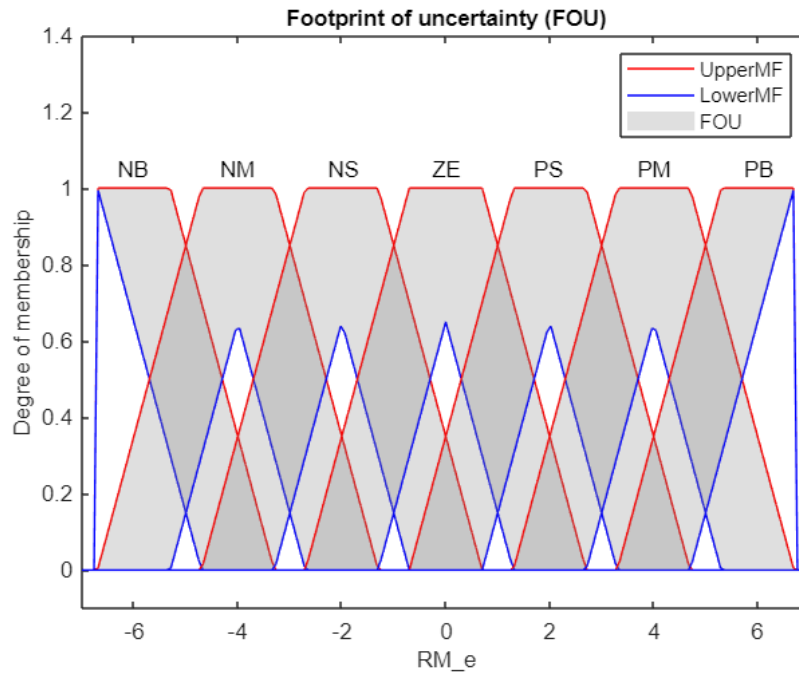
Regras_Fuzzy_T2_42 = zeros(6*25,5*2);
Regras_Fuzzy_T2_42(:,4:5) = 1;
Regras_Fuzzy_T2_42(:,9:10) = 1;
O11 = [4 2 1 1 , 3 1 1 , 4 2 1 1 , 3 1 1 , 3 1 1 1 , 3 1 1 , 3 1 1 1];
O12 = [4 3 1 1 , 3 1 1 , 4 2 1 1 , 3 1 1 , 4 2 1 1 , 2 1 1 , 4 2 1 1];
O13 = [4 2 1 1 , 3 1 1 , 4 2 1 1 , 3 1 1 , 4 2 1 1 , 3 1 1 , 4 2 1 1];
O14 = [4 2 1 1 , 3 1 1 , 4 2 1 1 , 3 1 1 , 3 1 1 1 , 3 1 1 , 3 1 1 1];

```

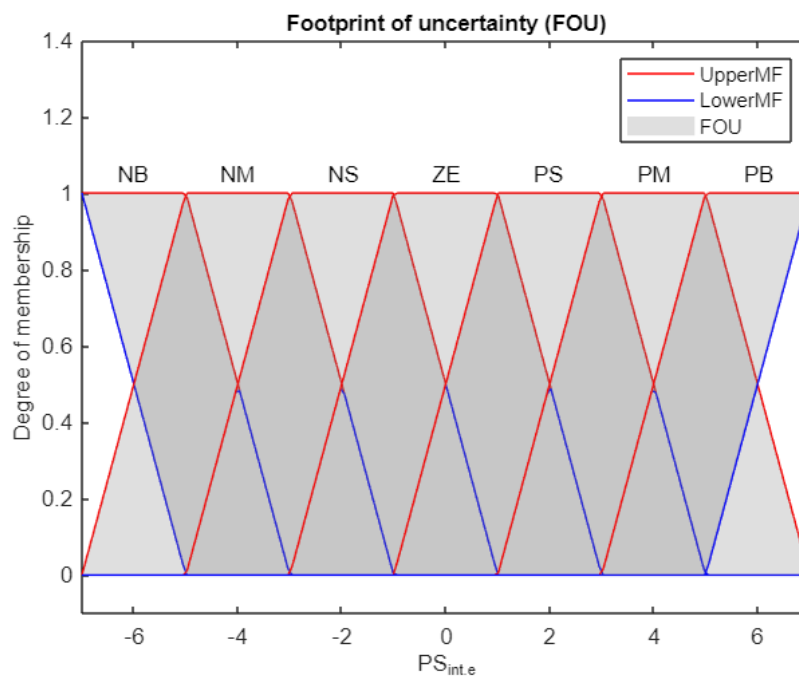
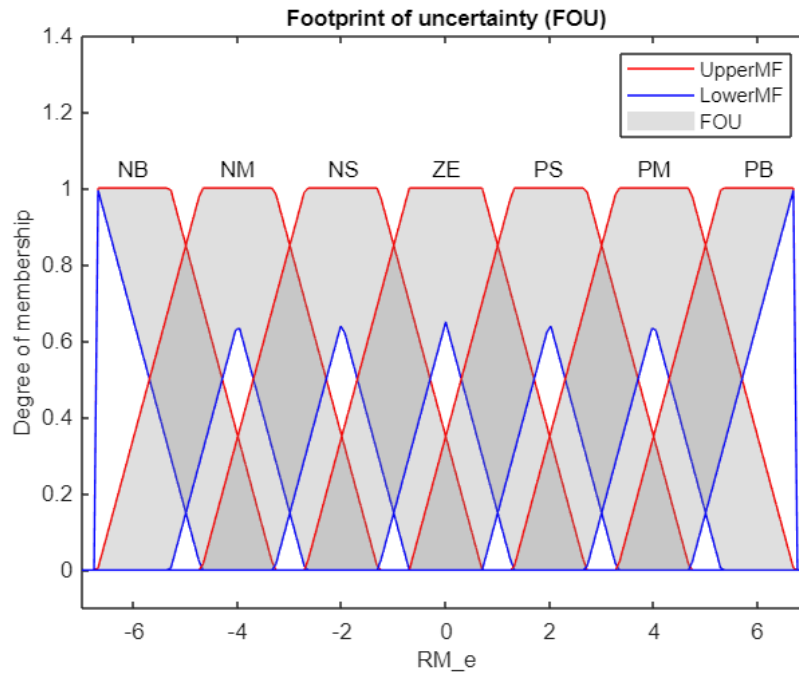
```

015 = [4 3 1 1 , 3 1 1 , 3 2 1 1 , 3 1 1 , 2 2 1 1 , 2 1 1 , 2 2 1 1];
016 = [1 1 1 1 , 1 1 1 , 1 1 1 1 , 1 1 1 , 1 1 1 1 , 1 1 1 , 1 1 1 1];
021 = [1 1 1 1 , 1 1 1 , 1 1 1 1 , 1 1 1 , 2 2 2 1 , 3 2 1 , 4 3 2 2];
022 = [1 1 1 1 , 1 1 1 , 1 1 1 1 , 1 1 1 , 2 2 2 1 , 3 2 1 , 4 3 2 2];
023 = [1 1 1 1 , 1 1 1 , 1 1 1 1 , 1 1 1 , 3 2 1 1 , 3 2 2 , 4 3 2 2];
024 = [1 1 1 1 , 1 1 1 , 1 1 1 1 , 1 1 1 , 2 2 2 1 , 3 2 1 , 4 3 2 2];
025 = [4 3 1 1 , 3 1 1 , 2 2 1 1 , 1 1 1 , 1 1 1 1 , 1 1 1 , 1 1 1 1];
026 = [1 1 1 2 , 1 1 2 , 1 1 2 3 , 1 1 3 , 1 2 2 4 , 2 3 4 , 2 3 4 4];
0 = [011,012,013,014,015,016;021,022,023,024,025,026];
cont1 = 1;
cont2 = 1;
for i = 1:6*25
    for j = 1:2
        d = 5*(j-1);
        Regras_Fuzzy_T2_42(i,1+d) = cont1;
        Regras_Fuzzy_T2_42(i,2+d) = cont2;
        Regras_Fuzzy_T2_42(i,3+d) = 0(j,i);
    end
    if cont1 == 7 && cont2 == 7
        cont1 = 1;
        cont2 = 1;
    end
    if cont1+2 <= 7
        cont1 = cont1 + 2;
    else
        if cont1+2 == 8
            cont1 = 1;
        else
            cont1 = 2;
        end
        cont2 = cont2 + 1;
    end
end
for i = 1:6
    for j = 1:2
        di1 = 25*(i-1) + 1;
        di2 = 25*i;
        dj1 = 5*(j-1) + 1;
        dj2 = 5*j;
        IT2_42_FLC_Anest(j,i) = addRule(IT2_42_FLC_Anest(j,i),...
            Regras_Fuzzy_T2_42(di1:di2,dj1:dj2));
    end
end
plotmf(IT2_42_FLC_Anest(1,1), 'input',1);
plotmf(IT2_42_FLC_Anest(1,1), 'input',2);

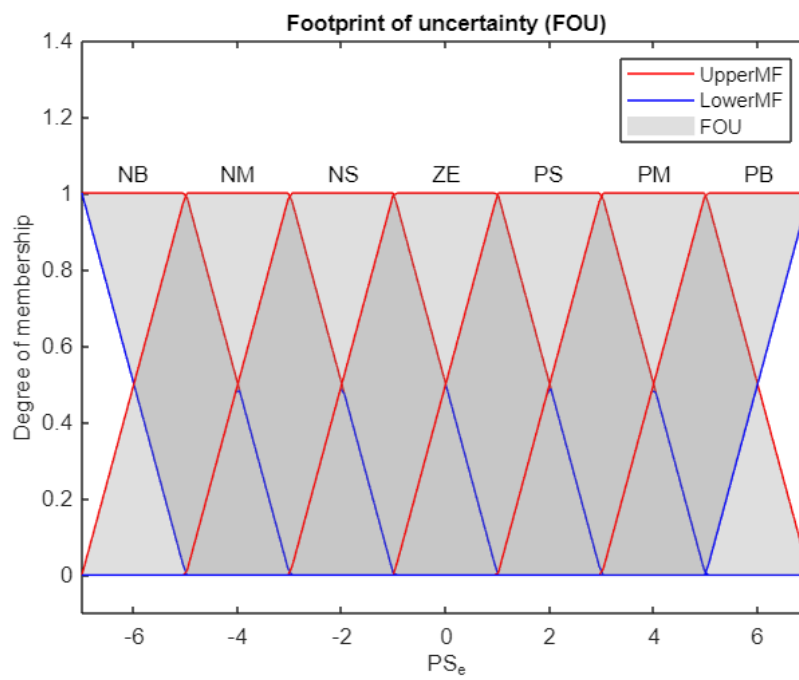
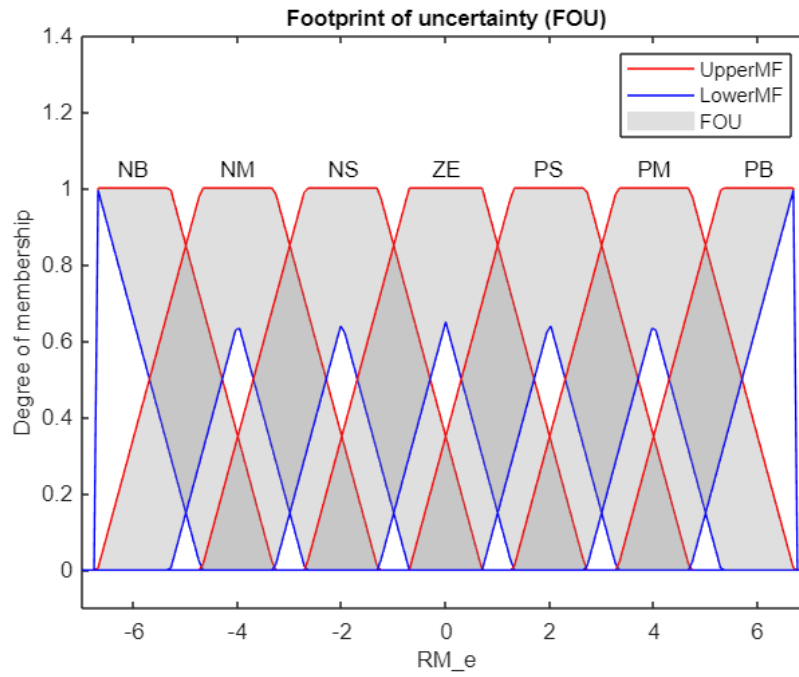
```



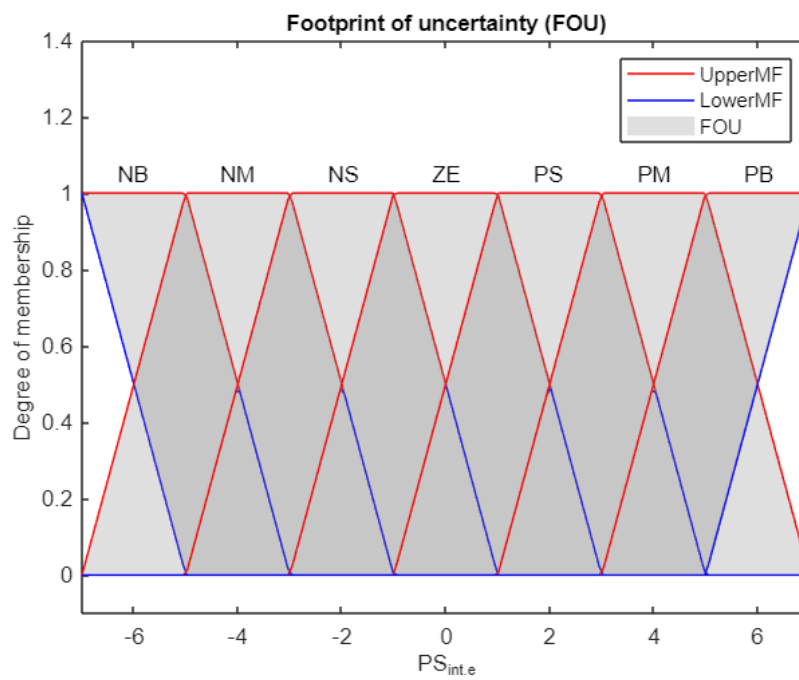
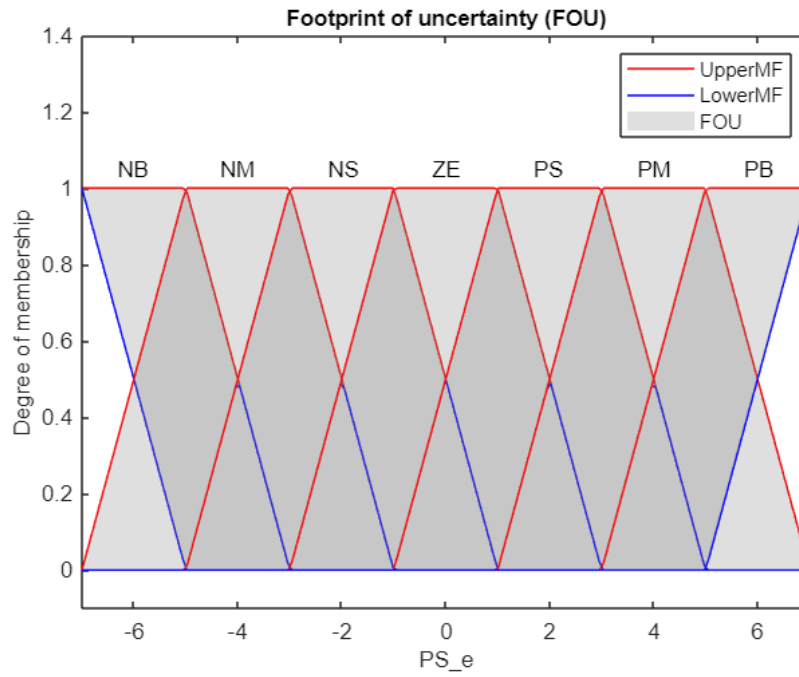
```
plotmf(IT2_42_FLC_Anest(1,3), 'input', 1);
plotmf(IT2_42_FLC_Anest(1,3), 'input', 2);
```



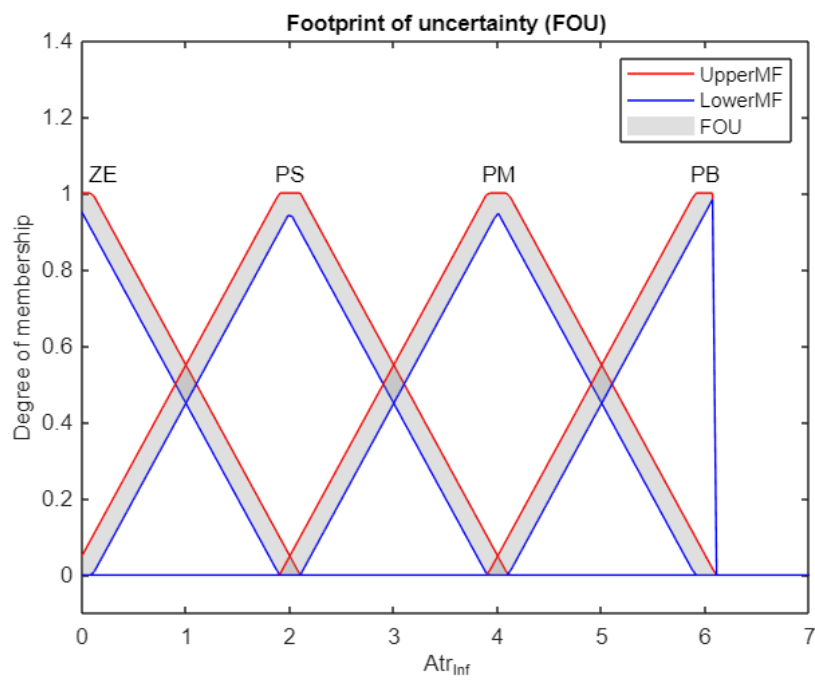
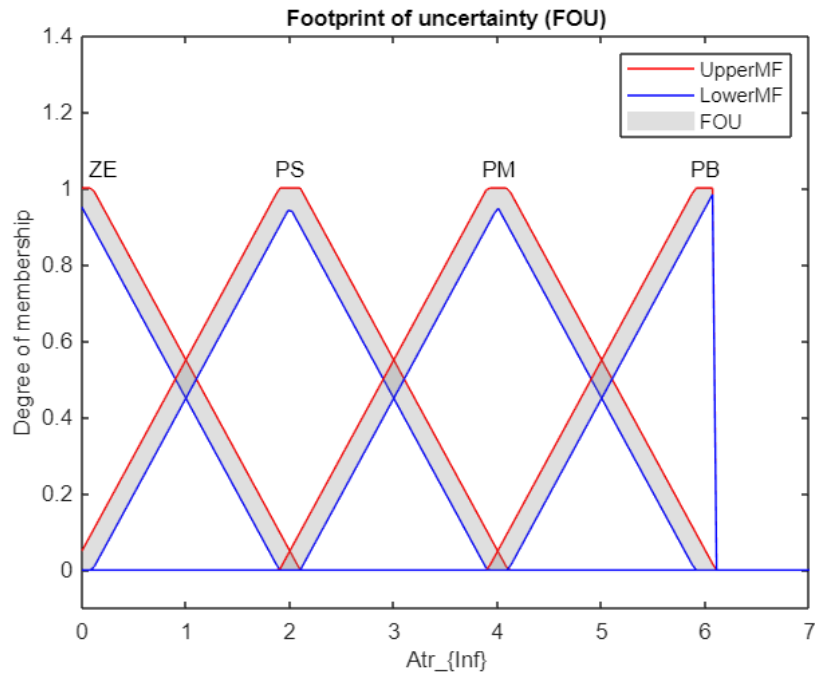
```
plotmf(IT2_42_FLC_Anest(2,2), 'input', 1);
plotmf(IT2_42_FLC_Anest(2,2), 'input', 2);
```



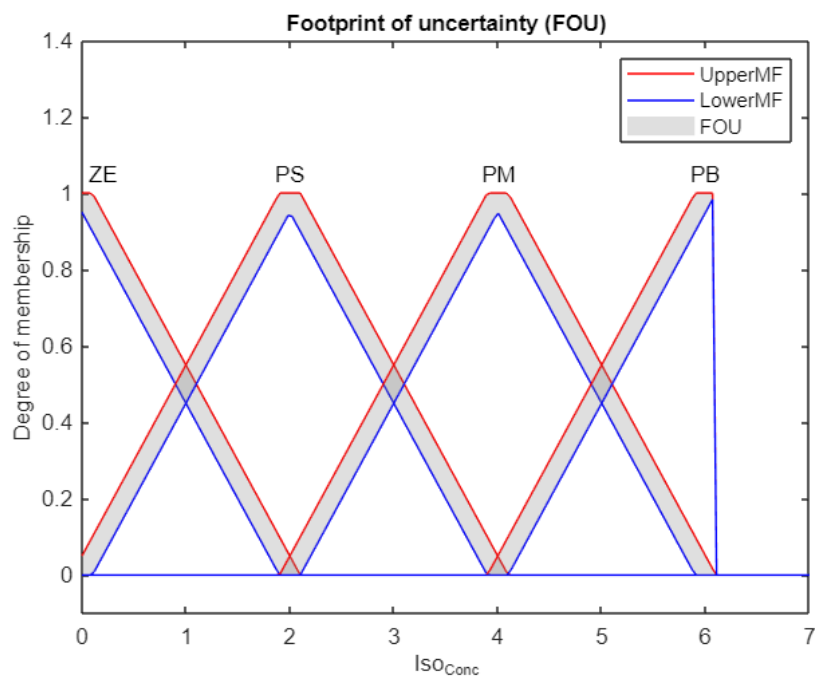
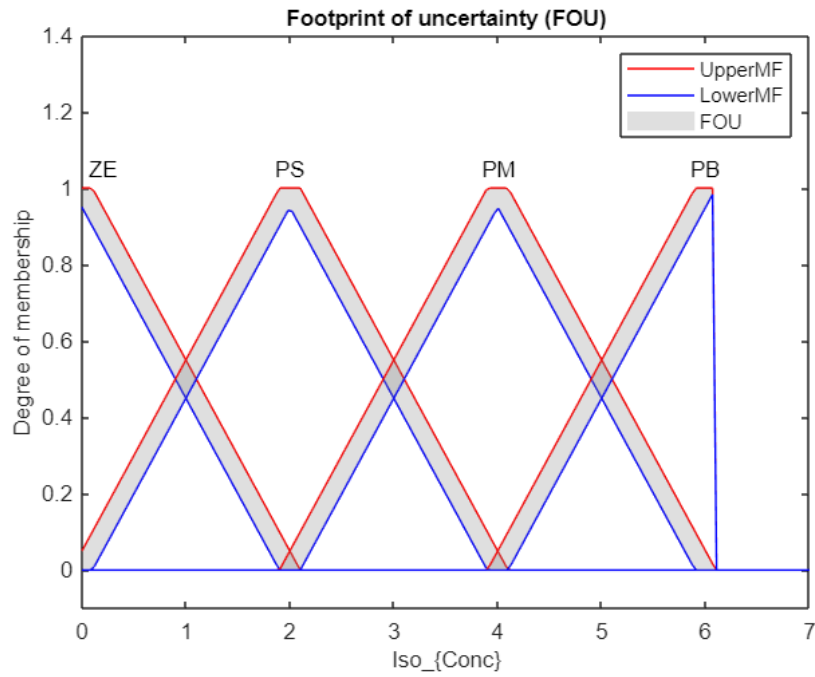
```
plotmf(IT2_42_FLC_Anest(2,6), 'input', 1);
plotmf(IT2_42_FLC_Anest(2,6), 'input', 2);
```



```
plotmf(IT2_42_FLC_Anest(1,1), 'output', 1);
plotmf(IT2_42_FLC_Anest(1,6), 'output', 1);
```



```
plotmf(IT2_42_FLC_Anest(2,1), 'output', 1);
plotmf(IT2_42_FLC_Anest(2,6), 'output', 1);
```



```
disp(IT2_42_FLC_Anest(1,1));
```

mamfistype2 with properties:

```
Name: "Controlador 11 IT2FLC"
AndMethod: "min"
OrMethod: "max"
ImplicationMethod: "min"
AggregationMethod: "max"
DefuzzificationMethod: "centroid"
```

```
        Inputs: [1x2 fisvar]
        Outputs: [1x1 fisvar]
        Rules: [1x25 fisrule]
DisableStructuralChecks: 0
    TypeReductionMethod: "eiasc"
```

See 'getTunableSettings' method for parameter optimization.

```
disp(IT2_42_FLC_Anest(1,6));
```

mamfistype2 with properties:

```
        Name: "Controlador 61 IT2FLC"
        AndMethod: "min"
        OrMethod: "max"
        ImplicationMethod: "min"
        AggregationMethod: "max"
        DefuzzificationMethod: "centroid"
        Inputs: [1x2 fisvar]
        Outputs: [1x1 fisvar]
        Rules: [1x25 fisrule]
DisableStructuralChecks: 0
    TypeReductionMethod: "eiasc"
```

See 'getTunableSettings' method for parameter optimization.

```
disp(IT2_42_FLC_Anest(2,1));
```

mamfistype2 with properties:

```
        Name: "Controlador 12 IT2FLC"
        AndMethod: "min"
        OrMethod: "max"
        ImplicationMethod: "min"
        AggregationMethod: "max"
        DefuzzificationMethod: "centroid"
        Inputs: [1x2 fisvar]
        Outputs: [1x1 fisvar]
        Rules: [1x25 fisrule]
DisableStructuralChecks: 0
    TypeReductionMethod: "eiasc"
```

See 'getTunableSettings' method for parameter optimization.

```
disp(IT2_42_FLC_Anest(2,6));
```

mamfistype2 with properties:

```
        Name: "Controlador 62 IT2FLC"  
        AndMethod: "min"  
        OrMethod: "max"  
        ImplicationMethod: "min"  
        AggregationMethod: "max"  
        DefuzzificationMethod: "centroid"  
        Inputs: [1x2 fisvar]  
        Outputs: [1x1 fisvar]  
        Rules: [1x25 fisrule]  
DisableStructuralChecks: 0  
TypeReductionMethod: "eiasc"
```

See 'getTunableSettings' method for parameter optimization.

```
save IT2_42_FLC.mat IT2_42_FLC_Anest;
```

APÊNDICE E – PROJETO DO MODELO PK/PD

Função para gerar o modelo PK-PD do Corpo Humano em Anestesia

Autor: Giovanni Oliveira de Sousa

Ano: 2021

O modelo PK-PD (Farmacocinética - Farmacodinâmica) do corpo humano em Anestesia (geral) é um modelo dinâmico não-linear que considera a interação geral multivariável entre as doses dos fármacos Atracúrio e Isoflurano relacionados ao efeito na **Pressão Sanguínea Arterial** e **Relaxamento muscular**.

```
clc
clear
close all
T = 0.01; % Período de discretização do modelo
```

1. ATRACÚRIO PARA O RELAXAMENTO MUSCULAR:

A. FARMACOCINÉTICA (PHARMACOKINETICS - PK)

De acordo com [1], a farmacocinética do Atracúrio, ou seja, a relação temporal entre a concentração do fármaco no sangue e a dose administrada, pode ser expressado pela função de transferência abaixo, onde $U(s)$ é a dose do fármaco e $X_1(s)$ é a concentração no compartimento 1 [2]:

$$\frac{X_1(s)}{U(s)} = G_1(s) = \frac{9.94(1 + 10.64s)}{(1 + 3.08s)(1 + 34.42s)}$$

B. FARMACODINÂMICA (PHARMACODYNAMICS - PD):

O modelo farmacodinâmico indica a relação da concentração do fármaco no sangue e seu efeito no corpo humano. Utilizando como referência [2], o modelo PK-PD linear do Atracúrio para o seu efeito no Relaxamento Muscular mostrará somente a relação entre a dose $U(s)$ e a concentração no compartimento Efeito $X_e(s)$ [2]:

$$\frac{X_e(s)}{U(s)} = G_{11}(s) = \frac{(1 + 10.64s)e^{-s}}{(1 + 3.08s)(1 + 34.42s)(1 + 4.81s)}$$

C. MODELO PK-PD NÃO-LINEAR:

Por último o modelo completo irá mostrar a relação entre a dose e o efeito, assim é necessário utilizar a equação de Hill [2] que torna o modelo não-linear:

$$E_{eff} = \frac{E_{max}}{1 + \frac{X_E(50)^\alpha}{X_E^\alpha}}$$

Os parâmetros da equação de Hill incluem o Efeito geral (Eeff), Efeito máximo (Emax), Concentração do fármaco (Xe), Concentração do fármaco para um efeito de 50% (Xe(50)) e uma constante α . Utilizando os valores de [1]:

$$E_{eff} = \frac{1}{1 + \frac{0.404^{2.98}}{(X_E)^{2.98}}}$$

O modelo final da dose Atracúrio para o efeito de Relaxamento Muscular é uma combinação das equações de PK-PD, ou seja, G11(s) e a equação de Hill para o Atracúrio. Essa equação não pode ser uma função de transferência, já que essa estrutura só aceita equações lineares. A seguir será gerado o modelo em linguagem MATLAB :

```
g11_num = [10.64 1];
g11_den = conv([3.08 1],conv([34.42 1],[4.81 1]));
G11_c = tf(g11_num,g11_den,'TimeUnit','minutes','IODelay',1);
G11_c % Função de transferência contínua
```

G11_c =

$$\exp(-1*s) * \frac{10.64 \text{ s} + 1}{509.9 \text{ s}^3 + 286.4 \text{ s}^2 + 42.31 \text{ s} + 1}$$

Continuous-time transfer function.

```
disp(G11_c);
```

tf with properties:

```

    Numerator: {[0 0 10.6400 1]}
  Denominator: {[509.9254 286.3886 42.3100 1]}
    Variable: 's'
      IODelay: 1
    InputDelay: 0
    OutputDelay: 0
         Ts: 0
    TimeUnit: 'minutes'
    InputName: {''}
    InputUnit: {''}
    InputGroup: [1x1 struct]
    OutputName: {''}
    OutputUnit: {''}
    OutputGroup: [1x1 struct]
        Notes: [0x1 string]
      UserData: []
         Name: ''
    SamplingGrid: [1x1 struct]
```

```
G11_d = c2d(G11_c,0.01);
G11_d % Função de transferência discretizada por T
```

```
G11_d =
```

```

      1.042e-06 z^2 - 6.439e-10 z - 1.039e-06
z^(-100) * -----
      z^3 - 2.994 z^2 + 2.989 z - 0.9944

```

```
Sample time: 0.01 minutes
```

```
Discrete-time transfer function.
```

```
disp(G11_d);
```

```
tf with properties:
```

```

      Numerator: {[0 1.0417e-06 -6.4395e-10 -1.0391e-06]}
      Denominator: {[1 -2.9944 2.9888 -0.9944]}
      Variable: 'z'
      IODelay: 100
      InputDelay: 0
      OutputDelay: 0
      Ts: 0.0100
      TimeUnit: 'minutes'
      InputName: {''}
      InputUnit: {''}
      InputGroup: [1x1 struct]
      OutputName: {''}
      OutputUnit: {''}
      OutputGroup: [1x1 struct]
      Notes: [0x1 string]
      UserData: []
      Name: ''
      SamplingGrid: [1x1 struct]

```

```

[g11_num_d,g11_den_d] = tfdata(G11_d,'v');
g11_delay = 1/T;
g11_XE_50 = 0.404;
g11_Alpha = 2.98;

```

2. ISOFLURANO PARA A PRESSÃO SANGUÍNEA ARTERIAL:

A. MODELO COMPLETO:

O modelo PK-PD do Isoflurano inalado pode ser aproximadamente linear para pequenas mudanças de concentração (menos de 5%) e é descrito em [1] como:

$$\frac{\Delta \text{MAP}(s)}{U_2(s)} = G_{22}(s) = \frac{-15e^{-0.42s}}{1 + 2s}$$

O modelo final da dose Isoflurano para o efeito de Pressão Sanguínea Arterial é linear com um atraso de transporte de 0,42 minuto. A seguir será gerado o modelo para a linguagem MATLAB:

```
g22_num = -15;
g22_den = [2 1];
G22_c = tf(g22_num,g22_den,'TimeUnit','minutes','IODelay',0.42);
G22_c % Função de transferência contínua
```

G22_c =

```

          -15
exp(-0.42*s) * -----
              2 s + 1
```

Continuous-time transfer function.

```
disp(G22_c);
```

tf with properties:

```

    Numerator: {[0 -15]}
  Denominator: {[2 1]}
    Variable: 's'
    IODelay: 0.4200
   InputDelay: 0
  OutputDelay: 0
         Ts: 0
    TimeUnit: 'minutes'
   InputName: {''}
  InputUnit: {''}
  InputGroup: [1x1 struct]
  OutputName: {''}
  OutputUnit: {''}
  OutputGroup: [1x1 struct]
        Notes: [0x1 string]
     UserData: []
         Name: ''
  SamplingGrid: [1x1 struct]
```

```
G22_d = c2d(G22_c,T,'Tustin');
G22_d % Função de transferência discretizada por T
```

G22_d =

$$z^{(-42)} * \frac{-0.03741 z - 0.03741}{z - 0.995}$$

Sample time: 0.01 minutes

Discrete-time transfer function.

```
disp(G22_d);
```

tf with properties:

```

    Numerator: {[-0.0374 -0.0374]}
  Denominator: {[1 -0.9950]}
    Variable: 'z'
      IODelay: 42
    InputDelay: 0
    OutputDelay: 0
          Ts: 0.0100
    TimeUnit: 'minutes'
    InputName: {''}
    InputUnit: {''}
    InputGroup: [1x1 struct]
    OutputName: {''}
    OutputUnit: {''}
    OutputGroup: [1x1 struct]
        Notes: [0x1 string]
      UserData: []
          Name: ''
    SamplingGrid: [1x1 struct]
```

```
[g22_num_d,g22_den_d] = tfdata(G22_d,'v');
g22_delay = 0.42/T;
```

3. ISOFLURANO PARA O RELAXAMENTO MUSCULAR

A. EFEITOS COLATERÁIS EM MODELOS MULTIVARIÁVEIS:

Cada fármaco é utilizado para um efeito específico na anestesia, porém, é possível que tenha um efeito não esperado em alguns casos. O Atracúrio não tem efeito significativo na Pressão Sanguínea Arterial [1], mas o Isoflurano tem um efeito no Relaxamento Muscular que precisa ser modelado. O efeito final no Relaxamento Muscular será a soma dos dois efeitos para cada fármaco.

B. MODELO COMPLETO:

De acordo com [1], o modelo do efeito do Isoflurano no Relaxamento Muscular é:

$$G_{12}(s) = \frac{0.27e^{-s}}{(1 + 2.83s)(1 + 1.25s)}$$

O modelo final da dose Isoflurano para o efeito de Relaxamento Muscular é linear com um atraso de transporte de 1 minuto. A seguir será gerado o modelo para a linguagem MATLAB:

```
g12_num = 0.27;
g12_den = conv([2.83 1],[1.25 1]);
G12_c = tf(g12_num,g12_den,'TimeUnit','minutes','IODelay',1);
G12_c % Função de transferência contínua
```

G12_c =

```

              0.27
exp(-1*s) *  -----
              3.538 s^2 + 4.08 s + 1
```

Continuous-time transfer function.

```
disp(G12_c);
```

tf with properties:

```

    Numerator: {[0 0 0.2700]}
  Denominator: {[3.5375 4.0800 1]}
    Variable: 's'
      IODelay: 1
    InputDelay: 0
    OutputDelay: 0
         Ts: 0
    TimeUnit: 'minutes'
   InputName: {''}
  InputUnit: {''}
 InputGroup: [1x1 struct]
 OutputName: {''}
 OutputUnit: {''}
 OutputGroup: [1x1 struct]
        Notes: [0x1 string]
      UserData: []
         Name: ''
 SamplingGrid: [1x1 struct]
```

```
G12_d = c2d(G12_c,T);
G12_d % Função de transferência discretizada por T
```

G12_d =

$$z^{(-100)} * \frac{3.802e-06 z + 3.787e-06}{z^2 - 1.989 z + 0.9885}$$

Sample time: 0.01 minutes

Discrete-time transfer function.

```
disp(G12_d);
```

tf with properties:

```

    Numerator: {[0 3.8016e-06 3.7870e-06]}
  Denominator: {[1 -1.9885 0.9885]}
    Variable: 'z'
      IODelay: 100
    InputDelay: 0
    OutputDelay: 0
          Ts: 0.0100
    TimeUnit: 'minutes'
    InputName: {''}
    InputUnit: {''}
    InputGroup: [1x1 struct]
    OutputName: {''}
    OutputUnit: {''}
    OutputGroup: [1x1 struct]
        Notes: [0x1 string]
      UserData: []
          Name: ''
    SamplingGrid: [1x1 struct]
```

```
[g12_num_d,g12_den_d] = tfdata(G12_d,'v');
g12_delay = 1/T;
```

4. MODELO MULTIVÁRIVEL DA ANESTESIA GERAL

Finalmente, o modelo multivárivel será a junção de todos os modelos para as entradas e saídas a seguir:

$RM(s)$ – Relaxamento Muscular;

$PS(s)$ – Pressão Sanguínea Arterial;

$U_1(s)$ – Dose de Atracúrio;

$U_2(s)$ – Dose de Isoflurano;

Uma maneira de visualizar o modelo é como uma matriz de Funções de Transferência [1]:

$$\begin{bmatrix} RM(s) \\ PS(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ 0 & G_{22}(s) \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix}$$

A simulação desse modelo será feita por meio de uma função criada do MATLAB, logo todas as informações pertinentes do modelo serão colocadas em somente uma matriz M:

```
g11_size = size(g11_num_d,2); % Ordem da função de transferência G11(z)
g12_size = size(g12_num_d,2); % Ordem da função de transferência G12(z)
g22_size = size(g22_num_d,2); % Ordem da função de transferência G22(z)
% Estrutura da Matriz geral da simulação do modelo:
% -> 3*2 linhas: São 3 Funções de transferência com 2 Vetores (num e
den)
% -> max(ordem das Funções de transferência)+1 colunas: Para caber cada
% coeficiente e o atraso de cada Função de transferência
M = zeros(3*2,max([g11_size g12_size g22_size])+1);
M_size = size(M,2)-1;
M(1,1:g11_size) = g11_num_d; % Coef. G11 numerador
M(2,1:g11_size) = g11_den_d; % Coef. G11 denominador
M(1,end) = g11_delay; % Coef. G11 atraso
M(2,end) = g11_size; % Coef. G11 ordem
M(3,1:g12_size) = g12_num_d; % Coef. G12 numerador
M(4,1:g12_size) = g12_den_d; % Coef. G12 denominador
M(3,end) = g12_delay; % Coef. G12 atraso
M(4,end) = g12_size; % Coef. G12 ordem
M(5,1:g22_size) = g22_num_d; % Coef. G22 numerador
M(6,1:g22_size) = g22_den_d; % Coef. G22 denominador
M(5,end) = g22_delay; % Coef. G22 atraso
M(6,end) = g22_size; % Coef. G22 ordem
```

Por último, os parâmetros mais importantes serão salvos em um arquivo .mat para serem utilizados na simulação.

```
save("PK_PD_Model_Parameters.mat", "g11_num_d", "g11_den_d", "g11_delay", "g
11_Alpha", "g11_XE_50", ...

"g12_num_d", "g12_den_d", "g12_delay", "g22_num_d", "g22_den_d", "g22_delay", "
M")
```

APÊNDICE F – FUNÇÃO DE SIMULAÇÃO DO CONTROLADOR T1.22.FLC

[illegible]

APÊNDICE G – FUNÇÃO DE SIMULAÇÃO DO CONTROLADOR T1.42.FLC

[illegible]

APÊNDICE H – FUNÇÃO DE SIMULAÇÃO DO CONTROLADOR IT2.22.FLC

[illegible]

APÊNDICE I – FUNÇÃO DE SIMULAÇÃO DO CONTROLADOR IT1.42.FLC

[illegible]

APÊNDICE J – FUNÇÃO DE SIMULAÇÃO DO MODELO PK/PD

```
% /////////////// Função para a Simulação do Modelo PK/PD \\\\\\\\\\\\\\\\\\\
% *Autor*: Giovanni Oliveira de Sousa
% *Ano*: 2021
% *Descrição*: Esse código serve como base da simulação do modelo PK/PD
% matemático dos remédios Atracúrio e Isoflurano no corpo humano para
% realizar a simulação do controle da anestesia. Os parâmetros utilizados
% foram descritos anteriormente no projeto do modelo PK/PD.
function [RM,PS,Y1,Y2,Y3] = PK_PD_Model_Sim(M,U,Y,i,XE_50,A,PS_ext)
    Dx = [M(1,end) M(3,end) M(5,end)]; % Vetor de Atraso
    Sx = [M(2,end) M(4,end) M(6,end)]; % Vetor de Ordem
    Ax = zeros(1,3); % Vetor de Diferença de Atraso
    Ox = [1,2,2]; % Vetor de Ordem MIMO
    Yx = zeros(1,3); % Vetor de Saída Auxiliar
    Numx = zeros(1,3); % Vetor de Soma Numerador
    Denx = zeros(1,3); % Vetor de Soma Denominador
    for j = 1:3
        if i > Dx(j)
            if i < Dx(j) + Sx(j)
                Ax(j) = Dx(j) + Sx(j) - i;
            end
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            for k = 1: Sx(j) - Ax(j)
                Numx(j) = Numx(j) + M(j+(j-1),k)*U(Ox(j),i-(k-1)-Dx(j));
                if k > 1
                    Denx(j) = Denx(j) + M(j+(j-1)+1,k)*Y(j,i-(k-1));
                end
            end
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            end
            Yx(j) = Numx(j) - Denx(j);
        end
        XE_num = (Yx(1)+Yx(2))^A; % Numerador do efeito XE
        XE_den = XE_num + XE_50^A; % Denominador do efeito XE
        RM = XE_num / XE_den; % Relaxamento Muscular
        PS = Yx(3) + PS_ext; % Pressão Sanguínea
        Y1 = Yx(1); % Saída G11
        Y2 = Yx(2); % Saída G12
        Y3 = Yx(3); % Saída G22
    end
end
```

APÊNDICE K – FUNÇÃO AUXILIAR DA SATURAÇÃO DAS ENTRADAS

```
% /////////////// Função para a Saturação de uma entrada \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
% *Autor*: Giovanni Oliveira de Sousa
% *Ano*: 2021
% *Descrição*: Esse código serve para manter o valor de uma entrada entre
% dois extremos, ou seja, passando desses, valor atribuído será o limite
% mais próximo.
function y = satur(u,min,max)
    if u > max
        y = max;
    elseif u < min
        y = min;
    else
        y = u;
    end
end
```

APÊNDICE L – FUNÇÃO AUXILIAR DO INTEGRADOR DAS ENTRADAS

```
% ////////////////////////////////// Função para integração numérica //////////////////////////////////
% *Autor*: Giovanni Oliveira de Sousa
% *Ano*: 2021
% *Descrição*: Esse código serve para integrar numericamente o valor das
% entradas utilizando o método trapezoidal, ou seja, será o acumulador da
% soma continua de valores.
function sum = integrator(u_n0,u_n1,T)
    % Trapezoidal
    sum = u_n0 + (T/2)*u_n1;
end
```

APÊNDICE M – T1.22.FLC: SIMULAÇÃO DO CONTROLE DA ANESTESIA

```
% ////////////// Teste de Simulação do Controlador T1.22.FLC \\\\\\\\\\\\\\\\\\\
% *Autor*: Giovanni Oliveira de Sousa
% *Ano*: 2021
% *Descrição*: Esse código serve para simular o controlador T1.22.FLC no
% modelo PK/PD do corpo humano na anestesia. São utilizados os arquivos de
% dados e funções customizadas para simular em 300 minutos com um período
% de 0,01 as respostas de RM e PS. Após o loop de simulação, o final do
% programa tem o objetivo de criar um gráfico com quatro janelas com as
% duas saídas e duas entradas simuladas.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Definir variáveis e carregar arquivos %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear
clear
addpath('T1_22_FLC_Aux_Functions_and_Data\');
load("T1_22_FLC.mat");
load("PK_PD_Model_Parameters.mat","M","g11_Alpha","g11_XE_50");
T = 0.01;
time = 300;
t = 0:T:time;
E = zeros(2,length(t)+1);
U = zeros(2,length(t));
Y = zeros(3,length(t)+1);
RM = zeros(1,length(t));
RM_sp = 0.8;
PS = zeros(1,length(t));
PS_sp = 110;
PS_ext = 120;
Fz_sc = [6/0.2 6/10 0.5 0.7];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Simulação em loop %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:length(t)
    if i == 1
        E(1,i) = RM_sp;
        E(2,i) = PS_sp - PS_ext;
    end
    [U(1,i),U(2,i)] =
    T1_22_FLC_Sim(E(1,i),E(2,i),Fz_sc,T1_22_FLC_Anest);
    [RM(i),PS(i),Y(1,i),Y(2,i),Y(3,i)] = PK_PD_Model_Sim(M,U,Y,i, ...
    g11_XE_50,g11_Alpha,PS_ext);
    E(1,i+1) = RM_sp - RM(1,i); % Malha Fechada
    E(2,i+1) = PS_sp - PS(1,i); % Malha Fechada
end
rmrpath('T1_22_FLC_Aux_Functions_and_Data\');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Gerar o gráfico com quatro janelas %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Sim = figure('Name','Simulação T1_22_FLC','NumberTitle','off');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Relaxamento Muscular %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(2,2,1);
plot(t,RM,'b',t,RM_sp*ones(1,length(t)),'k--');
title('Relaxamento Muscular');
```

```

ylabel('Relaxamento Muscular (norm.)');
xlabel('Tempo (min)');
ylim([0 1]);
grid on
legend('Simulação RM','Referência','Location','SouthEast');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Infusão Atracúrio %%%%%%%%%%%%%%
subplot(2,2,3);
plot(t,U(1,:));
title('Infusão de Atracúrio');
ylabel('Atracúrio (normalizado)');
xlabel('Tempo (min)');
grid on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Pressão Sanguínea %%%%%%%%%%%%%%
subplot(2,2,2);
plot(t,PS,'r',t,PS_sp*ones(1,length(t)),'k--');
title('Pressão Sanguínea Arterial');
ylabel('Pressão Sanguínea (mmHG)');
xlabel('Tempo (min)');
ylim([100 120]);
grid on
legend('Simulação PS','Referência','Location','SouthEast');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Concentração Isoflurano %%%%%%%%%%%%%%
subplot(2,2,4);
plot(t,U(2,:));
title('Concentração de Isoflurano');
ylabel('Isoflurano (normalizado)');
xlabel('Tempo (min)');
grid on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Salvar gráficos e gerar imagem %%%%%%%%%%%%%%
save T1_22_FLC_Anest_Sim.mat RM PS
exportgraphics(Sim,'T1_22_FLC_Simulacao.tif','Resolution',600);

```

APÊNDICE N – T1.42.FLC: SIMULAÇÃO DO CONTROLE DA ANESTESIA

[illegible]

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Sim = figure('Name','Simulação T1_42_FLC','NumberTitle','off');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Relaxamento Muscular %%%%%%%%%%%%%%%
subplot(2,2,1);
plot(t,RM,'b',t,RM_sp*ones(1,length(t)),'k--');
title('Relaxamento Muscular');
ylabel('Relaxamento Muscular (norm.)');
xlabel('Tempo (min)');
ylim([0 1]);
grid on
legend('Simulação RM','Referência','Location','SouthEast');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Infusão Atracúrio %%%%%%%%%%%%%%%
subplot(2,2,3);
plot(t,U(1,:));
title('Infusão de Atracúrio');
ylabel('Atracúrio (normalizado)');
xlabel('Tempo (min)');
grid on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Pressão Sanguínea %%%%%%%%%%%%%%%
subplot(2,2,2);
plot(t,PS,'r',t,PS_sp*ones(1,length(t)),'k--');
title('Pressão Sanguínea Arterial');
ylabel('Pressão Sanguínea (mmHG)');
xlabel('Tempo (min)');
ylim([100 120]);
grid on
legend('Simulação PS','Referência','Location','NorthEast');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Concentração Isoflurano %%%%%%%%%%%%%%%
subplot(2,2,4);
plot(t,U(2,:));
title('Concentração de Isoflurano');
ylabel('Isoflurano (normalizado)');
xlabel('Tempo (min)');
grid on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Salvar gráficos e gerar imagem %%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
save T1_42_FLC_Anest_Sim.mat RM PS
exportgraphics(Sim,'T1_42_FLC_Simulacao.tif','Resolution',600);

```

APÊNDICE O – IT2.22.FLC: SIMULAÇÃO DO CONTROLE DA ANESTESIA

```
% ////////////// Teste de Simulação do Controlador IT2.22.FLC \\\\\\\\\\\\\\\\\\\
% *Autor*: Giovanni Oliveira de Sousa
% *Ano*: 2021
% *Descrição*: Esse código serve para simular o controlador IT2.22.FLC no
% modelo PK/PD do corpo humano na anestesia. São utilizados arquivos de
% dados e funções customizadas para simular em 300 minutos com um período
% de 0,01 as respostas de RM e PS. Após o loop de simulação, o final do
% programa tem o objetivo de criar um gráfico com quatro janelas com as
% duas saídas e duas entradas simuladas.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Definir variáveis e carregar arquivos %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear
clear
addpath('IT2_22_FLC_Aux_Functions_and_Data\');
load('IT2_22_FLC.mat');
load('PK_PD_Model_Parameters.mat','M','g11_Alpha','g11_XE_50');
T = 0.01;
time = 300;
t = 0:T:time;
E = zeros(2,length(t)+1);
U = zeros(2,length(t));
Y = zeros(3,length(t)+1);
RM = zeros(1,length(t));
RM_sp = 0.8;
PS = zeros(1,length(t));
PS_sp = 110;
PS_ext = 120;
Fz_sc = [7/0.05 7/8 0.55 0.7];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Simulação em loop %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:length(t)
    if i == 1
        E(1,i) = RM_sp;
        E(2,i) = PS_sp - PS_ext;
    end
    [U(1,i),U(2,i)] =
    IT2_22_FLC_Sim(E(1,i),E(2,i),Fz_sc,IT2_22_FLC_Anest);
    [RM(i),PS(i),Y(1,i),Y(2,i),Y(3,i)] = PK_PD_Model_Sim(M,U,Y,i, ...
    g11_XE_50,g11_Alpha,PS_ext);
    E(1,i+1) = RM_sp - RM(1,i); % Malha Fechada
    E(2,i+1) = PS_sp - PS(1,i); % Malha Fechada
end
rmpath('IT2_22_FLC_Aux_Functions_and_Data\');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Gerar o gráfico com quatro janelas %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Sim = figure('Name','Simulação IT2_22_FLC','NumberTitle','off');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Relaxamento Muscular %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(2,2,1);
plot(t,RM,'b',t,RM_sp*ones(1,length(t)),'k--');
```

```

title('Relaxamento Muscular');
ylabel('Relaxamento Muscular (norm.)');
xlabel('Tempo (min)');
ylim([0 1]);
grid on
legend('Simulação RM','Referência','Location','SouthEast');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Infusão Atracúrio %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(2,2,3);
plot(t,U(1,:));
title('Infusão de Atracúrio');
ylabel('Atracúrio (normalizado)');
xlabel('Tempo (min)');
grid on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Pressão Sanguínea %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(2,2,2);
plot(t,PS,'r',t,PS_sp*ones(1,length(t)),'k--');
title('Pressão Sanguínea Arterial');
ylabel('Pressão Sanguínea (mmHG)');
xlabel('Tempo (min)');
ylim([100 120]);
grid on
legend('Simulação PS','Referência','Location','SouthEast');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Concentração Isoflurano %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(2,2,4);
plot(t,U(2,:));
title('Concentração de Isoflurano');
ylabel('Isoflurano (normalizado)');
xlabel('Tempo (min)');
grid on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Salvar gráficos e gerar imagem %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
save T2_22_FLC_Anest_Sim.mat RM PS
exportgraphics(Sim,'IT2_22_FLC_Simulacao.tif','Resolution',600);

```

APÊNDICE P – IT1.42.FLC: SIMULAÇÃO DO CONTROLE DA ANESTESIA

[illegible]

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Sim = figure('Name','Simulação IT2_42_FLC','NumberTitle','off');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Relaxamento Muscular %%%%%%%%%%%%%%%
subplot(2,2,1);
plot(t,RM,'b',t,RM_sp*ones(1,length(t)),'k--');
title('Relaxamento Muscular');
ylabel('Relaxamento Muscular (norm.)');
xlabel('Tempo (min)');
ylim([0 1]);
grid on
legend('Simulação RM','Referência','Location','SouthEast');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Infusão Atracúrio %%%%%%%%%%%%%%%
subplot(2,2,3);
plot(t,U(1,:));
title('Infusão de Atracúrio');
ylabel('Atracúrio (normalizado)');
xlabel('Tempo (min)');
grid on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Pressão Sanguínea %%%%%%%%%%%%%%%
subplot(2,2,2);
plot(t,PS,'r',t,PS_sp*ones(1,length(t)),'k--');
title('Pressão Sanguínea Arterial');
ylabel('Pressão Sanguínea (mmHG)');
xlabel('Tempo (min)');
ylim([100 120]);
grid on
legend('Simulação PS','Referência','Location','NorthEast');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Concentração Isoflurano %%%%%%%%%%%%%%%
subplot(2,2,4);
plot(t,U(2,:));
title('Concentração de Isoflurano');
ylabel('Isoflurano (normalizado)');
xlabel('Tempo (min)');
grid on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Salvar gráficos e gerar imagem %%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
save T2_42_FLC_Anest_Sim.mat RM PS
exportgraphics(Sim,'IT2_42_FLC_Simulacao.tif','Resolution',600);

```

APÊNDICE Q – COMPARAÇÃO ENTRE AS SIMULAÇÃO DE CONTROLE

```
% ////////////////////////////////////////////////// Comparação dos Resultados das Simulações //////////////////////////////////
% *Autor*: Giovanni Oliveira de Sousa
% *Ano*: 2021
% *Descrição*: Esse código serve para gerar dois gráficos que irão comparar
% os resultados das simulações dos controladores projetados para o controle
% da anestesia.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Definir variáveis e carregar arquivos %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear
addpath('T1_22_FLC\');
addpath('T1_42_FLC\');
addpath('T2_22_FLC\');
addpath('T2_42_FLC\');
auxL1 = load('T1_22_FLC_Anest_Sim.mat','RM','PS');
auxL2 = load('T1_42_FLC_Anest_Sim.mat','RM','PS');
auxL3 = load('T2_22_FLC_Anest_Sim.mat','RM','PS');
auxL4 = load('T2_42_FLC_Anest_Sim.mat','RM','PS');
rmpath('T1_22_FLC\');
rmpath('T1_42_FLC\');
rmpath('T2_22_FLC\');
rmpath('T2_42_FLC\');
T1_22_RM = auxL1.RM;
T1_22_PS = auxL1.PS;
T1_42_RM = auxL2.RM;
T1_42_PS = auxL2.PS;
T2_22_RM = auxL3.RM;
T2_22_PS = auxL3.PS;
T2_42_RM = auxL4.RM;
T2_42_PS = auxL4.PS;
t = 0:0.01:300;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Gerar gráficos %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Relaxamento Muscular %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RM = figure('Name','Comparação RM FLC','NumberTitle','off');
plot(t,T1_22_RM,'m',t,T1_42_RM,'b',t,T2_22_RM,'g',t,T2_42_RM,'r',...
'LineWidth',1.5);
hold on
plot(t,0.8*ones(1,length(t)),'k--');
hold off
title('Relaxamento Muscular');
ylabel('Relaxamento Muscular (normalizado)');
xlabel('Tempo (min)');
ylim([0 1]);
grid on
annotation('arrow',[0.13 0.23],[0.11 0.25]);
dimen_Z1 = [0.275,0.12,0.04,0.1];
annotation('textbox',dimen_Z1,'String','Zoom 1','FitBoxToText','on',...
'HorizontalAlignment','center');
annotation('arrow',[0.776 0.61],[0.76 0.82]);
dimen_Z2 = [0.71,0.8,0.09];
```

```

annotation('textbox',dimen_Z2,'String',"Zoom 2",'FitBoxToText','on',...
'HorizontalAlignment','center');
legend('T1.22.FLC','T1.42.FLC','IT2.22.FLC','IT2.42.FLC','Referência',...
'Location','Southeast','NumColumns',2);
axes('Position',[0.25 0.28 0.1 0.15],'Box','on');
RM_A = 100:230;
plot(t(RM_A),T1_22_RM(RM_A),'m',t(RM_A),T1_42_RM(RM_A),'b',t(RM_A),...
T2_22_RM(RM_A),'g',t(RM_A),T2_42_RM(RM_A),'r','LineWidth',1.5);
axes('Position',[0.5 0.82 0.11 0.08],'Box','on');
RM_B = 20000:25000;
plot(t(RM_B),T1_42_RM(RM_B),'b',t(RM_B),T2_42_RM(RM_B),'r',...
'LineWidth',1.5);
hold on
plot(t(RM_B),0.8*ones(1,length(t(RM_B))), 'k--');
plot(t(RM_B),0.82*ones(1,length(t(RM_B))), 'k');
plot(t(RM_B),0.79*ones(1,length(t(RM_B))), 'k');
hold off
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Pressão Sanguínea %%%%%%%%%%%%%%
PS = figure('Name','Comparação PS FLC','NumberTitle','off');
plot(t,T1_22_PS,'m',t,T1_42_PS,'b',t,T2_22_PS,'g',t,T2_42_PS,'r',...
'LineWidth',1.5);
hold on
plot(t,110*ones(1,length(t)), 'k--');
hold off
title('Pressão Sanguínea Arterial');
ylabel('Pressão Sanguínea (mmHG)');
xlabel('Tempo (min)');
ylim([100 120]);
grid on
annotation('arrow',[0.15 0.27],[0.53 0.46]);
dimen_Z3 = [0.436,0.32,0.04,0.1];
annotation('textbox',dimen_Z3,'String',"Zoom",'FitBoxToText','on',...
'HorizontalAlignment','center');
legend('T1.22.FLC','T1.42.FLC','IT2.22.FLC','IT2.42.FLC','Referência',...
'Location','Southeast','NumColumns',3);
axes('Position',[0.28 0.30 0.1 0.15],'Box','on');
PS_A = 400:1500;
plot(t(PS_A),T1_42_PS(PS_A),'b',t(PS_A),T2_42_PS(PS_A),'r',...
'LineWidth',1.5);
hold on
plot(t(PS_A),110*ones(1,length(t(PS_A))), 'k--');
plot(t(PS_A),109.5*ones(1,length(t(PS_A))), 'k');
hold off
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Salvar imagens %%%%%%%%%%%%%%
exportgraphics(RM,'RM_comparacao.tif','Resolution',600);
exportgraphics(RM,'RM_comparacao.jpg','Resolution',600);
exportgraphics(PS,'PS_comparacao.tif','Resolution',600);
exportgraphics(PS,'PS_comparacao.jpg','Resolution',600);

```