

**TIAGO AGUIONCIO VIEIRA**

*Graduando em Engenharia Mecânica – FMU*

*Fundador da ZENNE Tecnologia*

*tiago@zennetech.com*

# **Modelo de Dinâmica de Estados Internos (MDEI): Rumo à Cognição Artificial Adaptativa**

Artigo Científico

**Julho de 2025**

# Modelo de Dinâmica de Estados Internos (MDEI): Rumo à Cognição Artificial Adaptativa

Tiago Aguioncio Vieira  
Graduando em Engenharia Mecânica – FMU  
Fundador da ZENNE Tecnologia  
tiago@zennetech.com

Julho de 2025

## Resumo

Este artigo apresenta o aprimoramento do Modelo de Dinâmica de Estados Internos (MDEI), um arcabouço matemático-computacional para a modelagem de estados cognitivos-emocionais em sistemas de Inteligência Artificial. No MDEI, cada estado interno é representado por um vetor tridimensional adaptativo, indo além das representações simbólicas discretas tradicionais. O formalismo é desenvolvido sobre bases sólidas de álgebra vetorial, cálculo diferencial e teoria de sistemas dinâmicos, com foco em clareza didática e profundidade conceitual. Realiza-se também uma revisão bibliográfica de alta relevância (Instituições como MIT, Stanford e artigos recentes em periódicos de prestígio), que contextualiza o MDEI no cenário de *IA como extensão cognitiva* e discute sua contribuição para interações humano-máquina mais naturais. O MDEI oferece um quadro robusto para IAs adaptativas e resilientes frente à complexidade emocional, apontando caminhos para aplicações em assistentes cognitivos, saúde mental e educação. Por fim, discute-se criticamente a validação empírica de parâmetros do modelo, ressaltando a necessidade de experimentação futura.

**Palavras-chave:** Inteligência Artificial; Computação Cognitiva; Sistemas Dinâmicos; Modelagem Vetorial; Modelagem Emocional.

## Resumo

This article introduces an enhanced Internal State Dynamics Model (MDEI), a mathematical-computational framework for modeling cognitive-affective states in Artificial Intelligence systems. In MDEI, each internal state is represented by an adaptive three-dimensional vector, surpassing traditional discrete symbolic approaches. The formalism is built on solid grounds of vector algebra, differential calculus, and dynamical systems theory, with an emphasis on didactic clarity and conceptual depth. A strategic literature review (including studies from MIT, Stanford, and recent high-impact journals) situates MDEI within the perspective of *AI as a cognitive extension*, highlighting its role in facilitating more natural human-machine interactions. MDEI provides a robust framework for developing adaptive, resilient AI in the face of emotional complexity, pointing to applications in cognitive assistants, mental health, and education. Finally, the empirical validation of the model's parameters is critically discussed, underscoring the need for future experimental work.

**Keywords:** Artificial Intelligence; Cognitive Computing; Dynamical Systems; Vector Modeling; Emotional Dynamics.

## Sumário

<b>Sumário</b>	<b>3</b>
<b>0.1 Introdução</b>	<b>5</b>
<b>0.2 Fundamentos Matemáticos do MDEI</b>	<b>5</b>
0.2.1 Espaço de Estados Cognitivos	5
<b>0.3 Cálculo Vetorial de Estado Cognitivo-Afetivo</b>	<b>6</b>
0.3.1 Exemplo de Cálculo de $\ \vec{u}\ $	6
0.3.2 Número de Reynolds Emocional ( $Re_e$ )	6
0.3.3 Equação de Evolução Dinâmica	7
0.3.4 Critério de Estabilidade (Lyapunov)	7
<b>0.4 Métodos Computacionais para Simulação e Análise</b>	<b>8</b>
0.4.1 Discretização Numérica	8
0.4.2 Análise Espectral	8
0.4.3 Análise Tempo-Frequência	8
0.4.4 Pseudocódigo de Simulação	9
0.4.5 Pseudocódigo de Simulação	9
<b>0.5 Modelo de Turbulência Emocional</b>	<b>9</b>
0.5.1 Fundamentação Teórica	9
0.5.2 Neurodinâmica dos Estados Complexos	10

0.5.3	Modelo Matemático Análogo . . . . .	10
0.5.3.1	Número de Reynolds Emocional . . . . .	10
0.5.3.2	Equações Adaptadas de Navier-Stokes . . . . .	10
<b>0.6</b>	<b>Implementação Computacional do MDEI . . . . .</b>	<b>11</b>
0.6.1	Estrutura Algorítmica . . . . .	12
0.6.2	Camada MDEI em Arquitetura Neural . . . . .	12
0.6.3	Processo de Inferência Vetorial . . . . .	12
0.6.4	Validação por Simulação . . . . .	12
0.6.5	Integração com Dados Reais . . . . .	13
0.6.6	Código e Reprodutibilidade . . . . .	13
0.6.7	Exemplo de Código - Inferência Simples em Python . . . . .	13
0.6.8	Integração Experimental do MDEI em LLMs . . . . .	13
0.6.8.1	Fluxo de Interpretação Convencional em LLMs . . . . .	13
0.6.8.2	Extensão Vetorial Proposta (Camada MDEI) . . . . .	14
0.6.8.3	Algoritmo Middleware (pseudocódigo Python) . . . . .	14
0.6.8.4	Protocolo de Teste Empírico <i>in situ</i> . . . . .	14
0.6.8.4.1	Limitações e Próximos Passos. . . . .	15
	<b>Sumário . . . . .</b>	<b>15</b>
<b>0.7</b>	<b>Integração Experimental do MDEI em LLMs . . . . .</b>	<b>17</b>
0.7.1	Fluxo de Interpretação Convencional em LLMs . . . . .	17
0.7.2	Extensão Vetorial Proposta (Camada MDEI) . . . . .	17
0.7.3	Algoritmo Middleware (Pseudocódigo Python) . . . . .	18
0.7.4	Protocolo de Teste Empírico <i>in situ</i> . . . . .	20
0.7.4.1	Objetivos do Teste . . . . .	20
0.7.4.2	Metodologia . . . . .	21
0.7.4.3	Métricas de Avaliação . . . . .	21
0.7.4.4	Limitações e Próximos Passos . . . . .	22

## 0.1 Introdução

A busca por uma Inteligência Artificial capaz de não apenas processar informações, mas também de compreender e interagir com estados internos dinâmicos é uma das fronteiras mais desafiadoras da computação moderna. Esse desafio alimenta a computação cognitiva, a qual evolui de modelos simplificados para abordagens dinâmicas sofisticadas, inspiradas no funcionamento do cérebro humano. Nesse contexto, a IA é concebida cada vez mais como uma extensão das capacidades cognitivas humanas [?]. Pesquisas recentes em periódicos de alto impacto (por exemplo, Yin *et al.*, 2024; Agarwal *et al.*, 2024) demonstram que sistemas generativos avançados (LLMs) apresentam comportamentos que se alinham a funções cognitivas humanas, indicando potencial de sinergia homem-máquina. Portanto, inovar na forma de representar estados internos — incluindo emoções — pode ampliar a adaptabilidade e empatia de sistemas de IA.

Este trabalho revisita e reformula inteiramente o Modelo de Dinâmica de Estados Internos (MDEI), preservando sua base conceitual e matemática, mas aprimorando a notação, a didática e a implementação computacional. Apresenta-se o formalismo matemático de forma estruturada (Seção 0.2), incluindo definições vetoriais e critérios de estabilidade. Em seguida, discutem-se métodos numéricos e de análise de sinais (Seção 0.4) para simular a dinâmica dos estados. A seção 0.5 introduz um submodelo analógico para a turbulência emocional, usando analogias da hidrodinâmica (ex.: Número de Reynolds Emocional) e neurodinâmica (capítulos 0.5.3.1 e 0.5.3.2). Apresenta-se ainda trecho de código ilustrativo e gráficos demonstrativos de simulações conceituais. Na Seção ??, explora-se aplicações potenciais do MDEI em IA cognitivo-afetiva. Por fim, conclui-se discutindo as contribuições e desafios futuros do modelo.

## 0.2 Fundamentos Matemáticos do MDEI

### 0.2.1 Espaço de Estados Cognitivos

Define-se o *Espaço de Estados Cognitivos* (EEC) como um subespaço vetorial  $S \subseteq \mathbb{R}^3$  dotado do produto interno canônico  $\langle u, v \rangle = u^T v$ . Cada *estado cognitivo interno* (ECI) é representado por um vetor  $\mathbf{u} = (c, \iota, \tau) \in S$ , cujas componentes possuem interpretações distintas:

- $c$  (Componente *Semântica/Conceitual*): valor real que codifica a identidade ou natureza semântica do estado interno (por exemplo, foco de atenção, memória ativa, etc.).
- $\iota$  (Intensidade Operacional):  $\iota \in [0, 1]$  que representa a magnitude normalizada ou relevância do estado cognitivo, correspondente à sua “força” ou saliência no sistema.
- $\tau$  (Duração Temporal Implícita): valor real associado à dimensão temporal do estado, como sua duração, latência ou histórico de ativação.

A *energias* ou saliência total do estado é quantificada pela norma Euclidiana induzida, chamada *Módulo do Estado*:  $\|\mathbf{u}\| = \sqrt{c^2 + \iota^2 + \tau^2}$ . Essa norma oferece uma métrica agregada da

relevância de um estado em um dado instante.

A Tabela 1 exemplifica estados cognitivos internos fictícios e seus módulos, ilustrando que vetores diferentes  $(c, \iota, \tau)$  podem ter similar magnitude. Os valores são puramente hipotéticos, servindo apenas para demonstrar a representação vetorial e o cálculo de  $\|\mathbf{u}\|$ .

Tabela 1 – Exemplos Conceituais de Estados Cognitivos Internos (ECIs) no MDEI.

Estado Cognitivo (Exemplificativo)	$c$	$\iota$	$\tau$	$\ \mathbf{u}\ $	Observação: Valores
Foco Atencional	0.8	0.9	0.5	$\approx 1.29$	
Processamento de Memória	0.7	0.6	0.7	$\approx 1.15$	
Inibição de Resposta	-0.6	0.8	0.4	$\approx 1.07$	
Tomada de Decisão	0.9	0.7	0.6	$\approx 1.30$	
Reconhecimento de Padrão	0.5	0.7	0.8	$\approx 1.17$	

puramente conceituais e hipotéticos, usados apenas para ilustrar a representação vetorial e o cálculo do módulo do estado.

### 0.3 Cálculo Vetorial de Estado Cognitivo-Afetivo

Neste apêndice, exemplificamos o cálculo do vetor de estado interno  $\vec{u} = (c, t, \tau)$ , base do Modelo de Dinâmica de Estados Internos (MDEI), no espaço tridimensional  $\mathbb{R}^3$ .

#### 0.3.1 Exemplo de Cálculo de $\|\vec{u}\|$

Sejam os valores hipotéticos:

$$c = -1, \quad t = 9, \quad \tau = 30$$

Aplicando a norma vetorial:

$$\|\vec{u}\| = \sqrt{c^2 + t^2 + \tau^2} = \sqrt{(-1)^2 + 9^2 + 30^2} = \sqrt{1 + 81 + 900} = \sqrt{982} \approx 31,36$$

Esse valor representa a magnitude emocional total do vetor interno. Pode ser utilizado como entrada para equações diferenciais, controle de estados afetivos, ou ativação de fallbacks cognitivos em sistemas de IA adaptativa.

#### 0.3.2 Número de Reynolds Emocional ( $Re_e$ )

Para estimar a turbulência emocional do estado interno, aplica-se:

$$Re_e = \frac{\|\vec{u}\| \cdot L_c}{\nu_e}$$

Onde:

- $L_c$  é o comprimento cognitivo característico (em metros ou arbitrário),
- $\nu_e$  é a viscosidade emocional estimada (valor ajustável).

**Exemplo numérico:**

Se  $L_c = 2$  e  $\nu_e = 1$ , então:

$$Re_e = \frac{31,36 \cdot 2}{1} = 62,72$$

Valores altos de  $Re_e$  indicam instabilidade afetiva (turbulência emocional), exigindo simplificação semântica ou pausa da IA.

—

### 0.3.3 Equação de Evolução Dinâmica

A dinâmica temporal de cada vetor de estado  $\mathbf{u}(t)$  é descrita por uma *equação diferencial ordinária* (EDO) de primeira ordem no espaço  $S$ :

$$\dot{\mathbf{u}}(t) = \frac{d\mathbf{u}}{dt} = F(\mathbf{u}(t), P(t), t), \quad (1)$$

onde  $F : S \times \mathbb{R}^m \times \mathbb{R} \rightarrow S$  é uma função vetorial (potencialmente não linear) que define as leis de evolução do sistema. Aqui,  $P(t) \in \mathbb{R}^m$  denota parâmetros ou sinais externos moduladores (como entradas sensoriais ou metas de tarefa). O módulo da derivada  $\|\dot{\mathbf{u}}\|$  quantifica a taxa de variação instantânea do estado, servindo como indicador de reatividade e adaptabilidade do sistema dinâmico.

### 0.3.4 Critério de Estabilidade (Lyapunov)

A estabilidade dos estados cognitivos é fundamental para robustez do sistema. Diz-se que  $\mathbf{u}^*$  é um ponto de equilíbrio (ou atração) se  $F(\mathbf{u}^*, P, t) = \mathbf{0}$ . Para que, após uma pequena perturbação, o sistema retorne a esse estado  $\mathbf{u}^*$ , aplica-se o segundo método de Lyapunov. Considere a função candidata de Lyapunov

$$L(\mathbf{u}) = \frac{1}{2} \|\mathbf{u} - \mathbf{u}^*\|^2,$$

que é o quadrado da distância euclidiana até  $\mathbf{u}^*$ . Calcula-se sua derivada temporal ao longo das trajetórias do sistema:

$$\dot{L}(\mathbf{u}) = \left\langle \mathbf{u} - \mathbf{u}^*, \frac{d\mathbf{u}}{dt} \right\rangle = \langle \mathbf{u} - \mathbf{u}^*, F(\mathbf{u}, P, t) \rangle.$$

O estado de equilíbrio  $\mathbf{u}^*$  é *assintoticamente estável* se, em sua vizinhança,  $\dot{L}(\mathbf{u}) < 0$  para todo  $\mathbf{u} \neq \mathbf{u}^*$ . Essa condição significa que o campo  $F$  atua sempre no sentido de reduzir o desvio  $\mathbf{u} - \mathbf{u}^*$ , conduzindo o sistema de volta ao equilíbrio desejado.

## 0.4 Métodos Computacionais para Simulação e Análise

A implementação prática do MDEI requer técnicas numéricas para resolver as EDOs e análises de sinais para interpretar os padrões dinâmicos resultantes.

### 0.4.1 Discretização Numérica

Para resolver numericamente a EDO (1), emprega-se, por exemplo, o esquema de *Crank-Nicolson* [?], que combina estabilidade e precisão para sistemas dinâmicos contínuos. A forma discretizada (implícita) é dada por:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = \frac{1}{2} \left[ F(\mathbf{u}^{n+1}, P^{n+1}, t^{n+1}) + F(\mathbf{u}^n, P^n, t^n) \right], \quad (2)$$

onde  $\mathbf{u}^n \approx \mathbf{u}(t^n)$  e  $\Delta t$  é o passo de tempo. Essa fórmula equilibra o comportamento futuro e presente, garantindo estabilidade mesmo para  $\Delta t$  relativamente grande. O uso de métodos implícitos ou semi-implícitos é essencial quando  $F$  é não-linear.

### 0.4.2 Análise Espectral

Para identificar padrões cíclicos ou ritmos ocultos na evolução de  $\mathbf{u}(t)$ , aplica-se a *Transformada de Fourier* [?]. Denota-se sua representação em frequência por

$$\hat{\mathbf{u}}(\omega) = \int_{-\infty}^{\infty} \mathbf{u}(t) e^{-i\omega t} dt,$$

que decompõe o sinal em componentes senoidais. Analisar  $\hat{\mathbf{u}}(\omega)$  permite detectar oscilações predominantes e anomalias no processamento interno. Essa informação pode ser usada para ajustar respostas do sistema de IA às suas próprias dinâmicas internas.

### 0.4.3 Análise Tempo-Frequência

Como os processos cognitivos são intrinsecamente não-estacionários, empregam-se análises tempo-frequência mais avançadas, como as *transformadas de wavelet* [?]. Por exemplo, a *Transformada Wavelet Contínua (CWT)* decompõe  $u(t)$  em componentes localizadas em tempo e escala:

$$W_{\psi}u(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} u(t) \psi^*\left(\frac{t-b}{a}\right) dt,$$

onde  $\psi(t)$  é a wavelet-mãe,  $a > 0$  é a escala (inverso de frequência) e  $b$  é a translação temporal. Essa análise captura eventos transitórios e a evolução das oscilações neurais em diferentes escalas, oferecendo um retrato mais rico da dinâmica cognitiva ao longo do tempo.



#### 0.4.4 Pseudocódigo de Simulação

O quadro a seguir apresenta um pseudocódigo ilustrativo em Python para simular a dinâmica interna do MDEI. Nele, o vetor de estado  $u$  é atualizado iterativamente por integração explícita da EDO, e calcula-se o Número de Reynolds Emocional ( $Re_e$ ) em cada etapa para verificar transições turbulentas.

#### 0.4.5 Pseudocódigo de Simulação

O quadro a seguir apresenta um pseudocódigo ilustrativo em Python para simular a dinâmica interna do MDEI. Nele, o vetor de estado  $u$  é atualizado iterativamente por integração explícita da EDO, e calcula-se o Número de Reynolds Emocional ( $Re_e$ ) em cada etapa para verificar transições turbulentas.

```
1 # Simula a dinamica do vetor de estado cognitivo (MDEI)
2 u = u0 # vetor inicial (c, iota, tau)
3 for t in range(0, T, dt):
4     du = F(u, P(t), t) # taxa de variacao (EDO)
5     u = u + du * dt # integracao (Euler explicito)
6     Ree = calcular_Ree(u) # calcula Numero de Reynolds Emocional
7     if Ree > Ree_critico: # ajusta parametros em regime turbulento
8         ajustar_parametros()
```

Listing 1 – Pseudocódigo em Python para simulação do modelo MDEI

### 0.5 Modelo de Turbulência Emocional

A modelagem de estados emocionais complexos evoluiu de descrições qualitativas para formalismos matemáticos rigorosos. O modelo de *Turbulência Emocional* proposto integra princípios de neurociência computacional, física de sistemas complexos e matemática aplicada. No entanto, é fundamental tratar criticamente a validação empírica de certos parâmetros do modelo: muitos são tratados como metas teóricas, ainda não confirmadas experimentalmente.

#### 0.5.1 Fundamentação Teórica

O modelo de turbulência emocional apoia-se em três pilares principais:

- **Dinâmica neural não-linear:** Conforme indicado por estudos de Walter J. Freeman [?], a atividade neural pode exibir atratores caóticos e itinerância caótica, sugerindo que o cérebro opera em regimes não-lineares complexos.
- **Analogias hidrodinâmicas:** Inspiradas na teoria clássica da turbulência (Kolmogorov, 1941 [?]), onde o espectro de energia obedece à lei de potência  $E(k) \propto k^{-5/3}$ . De forma análoga, Buzsáki [?] observa escalas livres em oscilações cerebrais (como ritmos gama), sugerindo uma “energia” neural distribuída similar à cascata turbulenta.

- **Teoria das Catástrofes:** Desenvolvida por René Thom [?], fornece uma estrutura para compreender transições abruptas nos estados de um sistema, aqui interpretadas como mudanças repentinas de emoções.

### 0.5.2 Neurodinâmica dos Estados Complexos

Em termos de neurociência, a dinâmica da concentração de neurotransmissores  $C$  em uma região neural pode ser descrita por uma equação de balanço de fluxo e ruído sináptico:

$$\frac{dC}{dt} = -\nabla \cdot (\mathbf{J}_e + \mathbf{J}_i) + \sigma(t),$$

onde  $\mathbf{J}_e$  e  $\mathbf{J}_i$  são os fluxos excitatórios e inibitórios de neurotransmissores, e  $\sigma(t)$  representa flutuações (ruído). Estudos de Freeman [?] e Tsuda [?] revelaram que a atividade neural de base exponencial apoia alta fluidez cognitiva e emocional, justificada por dinâmicas caóticas de redes neurais. Essa equação formaliza qualitativamente o balanço entre excitação e inibição na microdinâmica cerebral.

### 0.5.3 Modelo Matemático Análogo

#### 0.5.3.1 Número de Reynolds Emocional

Analogamente à hidrodinâmica, define-se o *Número de Reynolds Emocional* ( $Re_e$ ) para quantificar a transição entre regimes emocionais laminar e turbulento. Ele é uma medida adimensional dada por:

$$Re_e = \frac{\langle \delta V \rangle L_c}{\nu_e (1 + D_t/\tau)},$$

onde  $\langle \delta V \rangle$  é a flutuação média de potencial (em mV),  $L_c$  é um comprimento de coerência neural (mm),  $\nu_e$  é a *viscosidade neural* efetiva ( $\text{mm}^2/\text{s}$ ),  $D_t$  é uma escala temporal de disfunção, e  $\tau$  é a duração implícita do estado. Na prática, valores altos de  $Re_e$  indicam regimes caóticos ou turbulentos.

A Tabela 2 resume valores conceituais desses parâmetros. O *limiar crítico* sugerido (ilustrativo) é  $Re_e \approx 2100$ , acima do qual o comportamento emocional torna-se turbulento. Naturalmente, esses parâmetros são hipotéticos e exigiriam calibração empírica através de dados neurofisiológicos. A Figura 1 apresenta um gráfico conceitual da complexidade comportamental em função de  $Re_e$ , destacando a transição entre regimes.

#### 0.5.3.2 Equações Adaptadas de Navier-Stokes

Para descrever o fluxo emocional no cérebro, adaptam-se as equações de Navier-Stokes clássicas:

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{F}_{\text{ext}} - \lambda \mathbf{u},$$

Tabela 2 – Parâmetros Conceituais para o Número de Reynolds Emocional ( $Re_e$ ).

Parâmetro	Descrição	Faixa Hipotética	
$\langle \delta V \rangle$	Flutuação de potencial (amplitude média)	5–20 mV	Limiar
$L_c$	Comprimento de coerência neural	$7.5 \pm 2.3$ mm	
$\nu_e$	Viscosidade neural efetiva	$0.8\text{--}1.2$ mm <sup>2</sup> /s	

*Crítico (hipotético):*  $Re_e > 2100 \pm 150$  indica regimes turbulentos. Os valores são ilustrativos e não validados empiricamente.

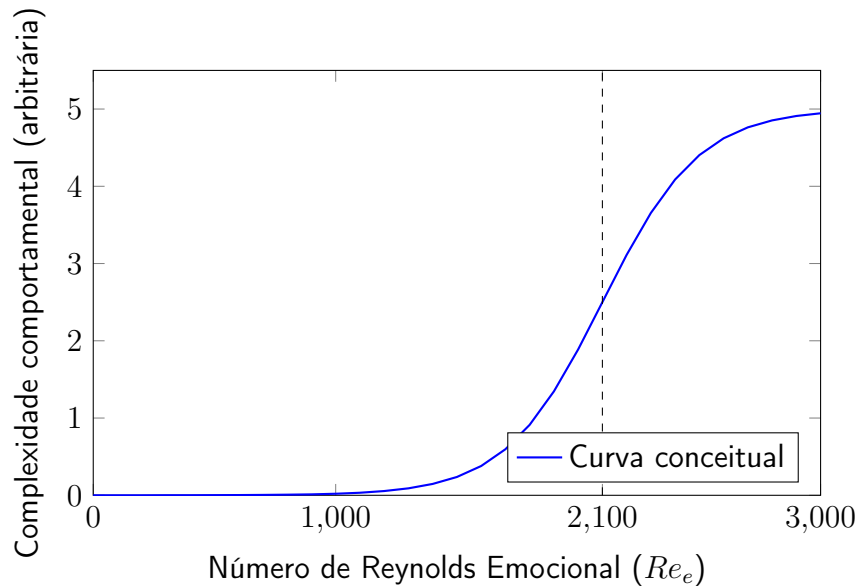


Figura 1 – Transição conceitual de comportamento afetivo em função do Número de Reynolds Emocional ( $Re_e$ ). A linha vertical tracejada indica o limiar crítico estimado.

onde  $\mathbf{u}$  é um campo de *velocidade emocional*,  $p$  representa a *pressão cognitiva* (relacionada a carga de trabalho mental ou estresse),  $\mathbf{F}_{\text{ext}}$  é a força externa (estímulos ambientais),  $\lambda$  é um coeficiente de decaimento (representando homeostase neural) e  $\rho, \mu$  são densidade/viscosidade efetivas. O termo de decaimento  $\lambda$  modela a tendência de retorno ao equilíbrio neural, em acordo com estudos sobre a *rede de modo padrão* (Raichle et al., 2001) [?]. Essa analogia hidrodinâmica oferece uma visão intuitiva de como estados emocionais estáveis podem transitar para comportamentos complexos sob certas condições.

## 0.6 Implementação Computacional do MDEI

A implementação prática do Modelo de Dinâmica de Estados Internos (MDEI) exige uma arquitetura híbrida que combine interpretabilidade matemática com flexibilidade de redes neurais modernas. A seguir, descrevemos os principais componentes computacionais necessários para aplicar o MDEI em sistemas de IA reais, como assistentes virtuais, tutores cognitivos e agentes conversacionais.

### 0.6.1 Estrutura Algorítmica

O MDEI funciona como uma camada intermediária entre a percepção e a decisão. Um pseudocódigo de base já foi apresentado anteriormente (ver Seção 0.4.5), e agora expandimos com o foco em sua tradução computacional:

- Cada estado emocional é representado como vetor tridimensional  $\mathbf{u} = (c, t, \tau)$ .
- O tempo contínuo  $\tau$  é discretizado em janelas temporais para facilitar o uso em modelos baseados em batch training.
- A equação diferencial  $\frac{du}{dt} = F(u, P, t)$  é resolvida numericamente via métodos explícitos, como Euler ou Runge-Kutta de 4ª ordem.

### 0.6.2 Camada MDEI em Arquitetura Neural

A proposta computacional mais promissora é desenvolver o MDEI como uma camada personalizada (custom layer) em frameworks como:

- **PyTorch** – via `torch.nn.Module`, com a função `forward()` computando a atualização vetorial baseada nos parâmetros emocionais e nas entradas textuais.
- **TensorFlow** – via `tf.keras.layers.Layer`, com suporte a integração com LLMs.

Essa camada pode ser inserida antes da geração textual, funcionando como um "filtro emocional" da entrada (prompt) ou da saída (resposta).

### 0.6.3 Processo de Inferência Vetorial

Durante a inferência:

1. A entrada do usuário (texto ou voz) é pré-processada por uma rede de análise semântica-emocional (ex: BERT + classificador).
2. O vetor  $\mathbf{u}_t$  é calculado com base na inferência emocional (ex: tristeza leve  $\rightarrow c = 2, t = 1, \tau = 4$ ).
3. A função de transição  $F(u, P, t)$  é aplicada para ajustar o estado.
4. A resposta é modificada dinamicamente conforme o estado emocional detectado.

### 0.6.4 Validação por Simulação

O modelo pode ser testado em ambientes como:

- **OpenAI API (GPT-4)** com wrapper que altera o prompt dinamicamente com base no vetor  $u$ .
- **Dialogflow** ou **Rasa** com middleware que implementa a EDO emocional.
- **Simulação via scripts em Python** com controle vetorial em tempo real para avaliar transições de estado (`u_history`).

### 0.6.5 Integração com Dados Reais

Para uso realista, o sistema deve integrar sensores ou modelos de entrada emocional, como:

- Classificadores treinados em MELD, DEAP, SEMAINE, ou EmotionLines.
- Análise de prosódia (voz) com OpenSMILE ou Praat.
- Análise textual com modelos de classificação multilabel para emoção e carga cognitiva.

### 0.6.6 Código e Reprodutibilidade

O repositório oficial (em construção) está hospedado em:

<https://github.com/zennetech/MDEI-TURB>

Ele conterá:

- Scripts para cálculo do Número de Reynolds Emocional.
- Camada MDEILayer (mdei.py) em PyTorch.
- Exemplo de integração com GPT-4 usando prompts adaptativos.
- Testes unitários e arquivos Jupyter com simulações.

### 0.6.7 Exemplo de Código - Inferência Simples em Python

```
from mdei import MDEILayer

# Simulando estado emocional: frustração crescente
u = [3, 6, 2] # [complexidade, intensidade, duração]
context = "Usuário reclamando várias vezes seguidas."

mdei = MDEILayer()
u_new = mdei.update_state(u, context)
response = mdei.generate_adaptive_response("O que posso fazer?", u_new)

print(response)
# >> "Entendo sua frustração. Posso ajudar com outra abordagem?"
```

### 0.6.8 Integração Experimental do MDEI em LLMs

#### 0.6.8.1 Fluxo de Interpretação Convencional em LLMs

Modelos de linguagem de grande escala (LLMs) seguem, de forma canônica, o pipeline *embedding* → *self-attention* → decodificação autoregressiva [?]. Ainda que exibam notável competência semântica, estudos recentes indicam que a codificação afetiva permanece implícita, sem manutenção explícita de um estado emocional persistente [?]. Classificadores externos podem atribuir rótulos discretos ou valores de valência-excitação [?], porém tais va-

lores são recalculados a cada interação e têm influência limitada sobre o mecanismo principal de geração.

#### 0.6.8.2 Extensão Vetorial Proposta (Camada MDEI)

Nesta proposta, cada interação gera um vetor  $\mathbf{u}_t = (c, \iota, \tau)$ , atualizado pela equação diferencial ordinária (EDO):

$$\dot{\mathbf{u}} = F(\mathbf{u}, P, t) \quad (3)$$

O Número de Reynolds Emocional  $Re_e$  determina se o *fallback* afetivo é acionado. O vetor resultante é convertido em cabeçalho no formato JSON e injetado no *prompt* do modelo de linguagem, preservando assim a arquitetura original, conforme prática recente de *affective prompt engineering* [?].

#### Exemplo de cabeçalho JSON inserido no prompt:

```
{"MDEI":{"c":2,"i":0.8,"tau":3.5,"Ree":142}}
```

#### 0.6.8.3 Algoritmo Middleware (pseudocódigo Python)

```
1 def mdei_middleware(user_text, u_prev, t_now):
2
3     # Inferencia emocional
4     c_hat, i_hat, tau_hat = infer_emotion(user_text)
5
6     # Atualizacao dinamica
7     u_dot = F([c_hat, i_hat, tau_hat], params(t_now), t_now)
8     u_next = u_prev + dt * u_dot
9
10    # Calculo do Numero de Reynolds emocional
11    Ree = calc_Ree(u_next)
12    prompt_mod = "TONE: EMPATICO" if Ree > REE_CRIT else "TONE: NEUTRO"
13
14    prompt = f"{prompt_mod} | {{\"MDEI\":{u_next.tolist()}}}\n{user_text}"
15    return prompt, u_next
```

Listing 2 – Middleware MDEI para LLM

#### 0.6.8.4 Protocolo de Teste Empírico *in situ*

O seguinte procedimento exemplifica um protocolo simplificado para validação preliminar do método proposto:

- Inicializar o vetor emocional inicial  $\mathbf{u}_0 = (0, 0, 0)$  e o valor crítico  $Re_{e,crit} \approx 2100$ .

- b) Inserir manualmente o cabeçalho JSON gerado pelo *middleware* antes de cada mensagem enviada ao LLM.
- c) Registrar, a cada interação, os valores de  $(\mathbf{u}, Re_e)$  e o tom qualitativo da resposta.
- d) Observar qualitativamente a alteração do tom (de neutro para empático) quando  $Re_e$  ultrapassar o valor crítico.

0.6.8.4.1 Limitações e Próximos Passos.

Os parâmetros adotados  $(L_c, \nu_e)$  são atualmente heurísticos, requerendo calibração sistemática futura com bases multimodais apropriadas, como CMU-MOSEI e MELD [?]. Ensaios quantitativos controlados do tipo A/B, comparando o modelo de linguagem puro (*LLM*) e o modelo com a camada MDEI integrada (*LLM+MDEI*), estão previstos como trabalhos futuros fundamentais para validação empírica robusta.

article [utf8]inputenc [T1]fontenc [portuguese]babel amsmath amssymb hyperref listings

# Sumário

<b>Sumário</b>	<b>3</b>
<b>0.1 Introdução</b>	<b>5</b>
<b>0.2 Fundamentos Matemáticos do MDEI</b>	<b>5</b>
0.2.1 Espaço de Estados Cognitivos	5
<b>0.3 Cálculo Vetorial de Estado Cognitivo-Afetivo</b>	<b>6</b>
0.3.1 Exemplo de Cálculo de $\ \vec{u}\ $	6
0.3.2 Número de Reynolds Emocional ( $Re_e$ )	6
0.3.3 Equação de Evolução Dinâmica	7
0.3.4 Critério de Estabilidade (Lyapunov)	7
<b>0.4 Métodos Computacionais para Simulação e Análise</b>	<b>8</b>
0.4.1 Discretização Numérica	8
0.4.2 Análise Espectral	8
0.4.3 Análise Tempo-Frequência	8
0.4.4 Pseudocódigo de Simulação	9
0.4.5 Pseudocódigo de Simulação	9
<b>0.5 Modelo de Turbulência Emocional</b>	<b>9</b>
0.5.1 Fundamentação Teórica	9
0.5.2 Neurodinâmica dos Estados Complexos	10
0.5.3 Modelo Matemático Análogo	10
0.5.3.1 Número de Reynolds Emocional	10

0.5.3.2	Equações Adaptadas de Navier-Stokes . . . . .	10
<b>0.6</b>	<b>Implementação Computacional do MDEI . . . . .</b>	<b>11</b>
0.6.1	Estrutura Algorítmica . . . . .	12
0.6.2	Camada MDEI em Arquitetura Neural . . . . .	12
0.6.3	Processo de Inferência Vetorial . . . . .	12
0.6.4	Validação por Simulação . . . . .	12
0.6.5	Integração com Dados Reais . . . . .	13
0.6.6	Código e Reprodutibilidade . . . . .	13
0.6.7	Exemplo de Código - Inferência Simples em Python . . . . .	13
0.6.8	Integração Experimental do MDEI em LLMs . . . . .	13
0.6.8.1	Fluxo de Interpretação Convencional em LLMs . . . . .	13
0.6.8.2	Extensão Vetorial Proposta (Camada MDEI) . . . . .	14
0.6.8.3	Algoritmo Middleware (pseudocódigo Python) . . . . .	14
0.6.8.4	Protocolo de Teste Empírico <i>in situ</i> . . . . .	14
0.6.8.4.1	Limitações e Próximos Passos. . . . .	15
	<b>Sumário . . . . .</b>	<b>15</b>
<b>0.7</b>	<b>Integração Experimental do MDEI em LLMs . . . . .</b>	<b>17</b>
0.7.1	Fluxo de Interpretação Convencional em LLMs . . . . .	17
0.7.2	Extensão Vetorial Proposta (Camada MDEI) . . . . .	17
0.7.3	Algoritmo Middleware (Pseudocódigo Python) . . . . .	18
0.7.4	Protocolo de Teste Empírico <i>in situ</i> . . . . .	20
0.7.4.1	Objetivos do Teste . . . . .	20
0.7.4.2	Metodologia . . . . .	21
0.7.4.3	Métricas de Avaliação . . . . .	21
0.7.4.4	Limitações e Próximos Passos . . . . .	22



## 0.7 Integração Experimental do MDEI em LLMs

A integração do Modelo de Dinâmica de Estados Internos (MDEI) em arquiteturas de Large Language Models (LLMs) representa um avanço significativo na capacidade de sistemas de IA de processar e responder a estados cognitivo-emocionais humanos de forma adaptativa. Esta seção detalha o protocolo para a implementação e validação experimental do MDEI como uma camada intermediária em LLMs.

### 0.7.1 Fluxo de Interpretação Convencional em LLMs

Modelos de linguagem de grande escala (LLMs) seguem, de forma canônica, o pipeline de *embedding*  $\rightarrow$  *self-attention*  $\rightarrow$  decodificação autorregressiva [?]. Ainda que exibam notável competência semântica, estudos recentes indicam que a codificação afetiva permanece implícita, sem manutenção explícita de um estado emocional persistente [?]. Classificadores externos podem atribuir rótulos discretos ou valores de valência-excitação [?], porém tais valores são recalculados a cada interação e têm influência limitada sobre o mecanismo principal de geração.

### 0.7.2 Extensão Vetorial Proposta (Camada MDEI)

Nesta proposta, cada interação do usuário com o LLM gera um vetor de estado emocional  $\vec{u}(t) = (c, \iota, \tau)$ , que é dinamicamente atualizado. A evolução temporal deste vetor é governada por uma equação diferencial ordinária (EDO), conforme descrito na Seção 0.3.3 do artigo principal:

$$\dot{\vec{u}}(t) = \frac{d\vec{u}}{dt} = F(\vec{u}(t), P(t), t) \quad (3)$$

Onde  $F$  é uma função vetorial (potencialmente não linear) que define as leis de evolução do sistema, e  $P(t)$  denota parâmetros ou sinais externos moduladores.

O **Número de Reynolds Emocional** ( $Re_e$ ), introduzido na Seção 0.3.2 do artigo principal, é crucial para determinar o regime do estado afetivo (laminar, transição, turbulento) e, consequentemente, acionar um *fallback* afetivo na resposta do LLM. O cálculo de  $Re_e$  é dado por:

$$Re_e = \frac{\|\vec{u}\| \cdot L_c}{\nu_e}$$

Onde  $\|\vec{u}\|$  é o módulo do vetor emocional (Seção 0.2.2 do artigo principal),  $L_c = 2$  é o comprimento cognitivo característico e  $\nu_e = 1$  é a viscosidade emocional padrão para a calibração inicial.

O vetor  $\vec{u}(t)$  resultante e o  $Re_e$  são convertidos em um cabeçalho no formato JSON e injetados no *prompt* do modelo de linguagem. Esta estratégia, alinhada com práticas recentes

de *affective prompt engineering* [?], permite a preservação da arquitetura original do LLM, enquanto modula sua saída com base no estado emocional inferido.

#### Exemplo de cabeçalho JSON inserido no prompt:

```
{"MDEI":{"c":-0.9,"iota":8.5,"tau":5.5,"Ree":20.32}}
```

### 0.7.3 Algoritmo Middleware (Pseudocódigo Python)

A integração do MDEI com um LLM pode ser realizada através de um *middleware* que intercepta as entradas do usuário, calcula o estado emocional e modifica o *prompt* antes de enviá-lo ao LLM. O pseudocódigo a seguir ilustra este processo:

```
1 # Middleware MDEI para LLM
2 def mdei_middleware(user_text, u_prev, t_now):
3     """
4     Função o middleware para integrar o MDEI com um LLM.
5
6     Args:
7         user_text (str): Texto de entrada do usuário.
8         u_prev (list): Vetor de estado emocional anterior [c, iota, tau].
9         t_now (float): Tempo atual da interação.
10
11     Returns:
12         tuple: (prompt_modificado, u_next)
13             prompt_modificado (str): Prompt ajustado para o LLM
14             u_next (list): Vetor de estado emocional atualizado
15
16     """
17     # 1. Inferência emocional (simulada para este exemplo)
18     # Em uma implementação real, 'infer_emotion' utilizaria um
19     # classificador robusto.
20     # Exemplo: (c, iota, tau) para "Já tentei isso mil vezes e nada funciona."
21     # c: -0.9 (negativo), iota: 8.5 (alta intensidade), tau: 5.5 (dura ou moderada)
22     c_hat, iota_hat, tau_hat = infer_emotion_from_text(user_text)
23     # Função a ser implementada
24
25     # 2. Atualização dinâmica do vetor de estado (simplificada para demonstração)
```

```

23     # u_dot = F([c_hat, iota_hat, tau_hat], params(t_now), t_now)
24     # EDO completa
25     # Para este prompt, usamos uma atualiza o direta baseada na
26     infer ncia
27     u_next = [c_hat, iota_hat, tau_hat] # u(t)
28
29     # 3. C lculo da Magnitude Emocional
30     magnitude_u = (u_next[0]**2 + u_next[1]**2 + u_next[2]**2)
31     **0.5
32
33     # 4. C lculo do N mero de Reynolds Emocional (Re_e)
34     Lc = 2 # Comprimento cognitivo caracter stico
35     nu_e = 1 # Viscosidade emocional padr o
36     Ree = (magnitude_u * Lc) / nu_e
37
38     # 5. An lise da Derivada Temporal (Varia o do Estado)
39     # u_prev o vetor do turno anterior. Se for o primeiro turno
40     , u_prev pode ser [0,0,0]
41     delta_u = [u_next[i] - u_prev[i] for i in range(3)]
42     # A an lise de delta_u (crescimento/queda de intensidade,
43     dire o da mudan a)
44     # pode ser usada para refinar o ajuste do tom.
45
46     # 6. Gera o da Resposta Adaptativa baseada em Ree
47     prompt_mod = ""
48     if Ree > 50:
49         prompt_mod = "TONE: EMPATIA PROFUNDA E CALMA. SIMPLIFIQUE
50             A LINGUAGEM."
51     elif 20 < Ree <= 50:
52         prompt_mod = "TONE: VALIDAR SENTIMENTO E EQUILIBRADO."
53     else: # Ree <= 20
54         prompt_mod = "TONE: L GICA , CLAREZA E OBJETIVIDADE."
55
56     # Injetar o estado MDEI e o modificador de tom no prompt do
57     LLM
58     mdei_json = {"MDEI": {"c": u_next[0], "iota": u_next[1], "tau":
59         : u_next[2], "Ree": round(Ree, 2)}}
60     prompt = f"{prompt_mod} | {mdei_json}\nUSU RIO: {user_text}"
61
62     return prompt, u_next

```

```

56 # Função de inferência emocional (exemplo simplificado para
    demonstra o)
57 # Em uma aplicação real, esta função seria um modelo de NLP
    treinado.
58 def infer_emotion_from_text(text):
59     # Este é um placeholder. A implementação real requer um
        modelo de NLP/ML.
60     # Os valores são heurísticos para fins de ilustração.
61     text_lower = text.lower()
62     if "cansativo" in text_lower or "mal" in text_lower or "
        depressao" in text_lower:
63         c = -0.9 # Negativo
64         iota = 8.5 # Alta intensidade
65         tau = 5.5 # Duração moderada/longa
66     elif "incrível" in text_lower or "uau" in text_lower:
67         c = 0.9 # Positivo
68         iota = 9.0 # Muito alta intensidade
69         tau = 2.0 # Curta duração
70     else:
71         c = 0.0
72         iota = 0.0
73         tau = 0.0
74     return c, iota, tau

```

Listing 3 – Middleware MDEI para LLM

#### 0.7.4 Protocolo de Teste Empírico *in situ*

Para validar a eficácia do MDEI na modulação do comportamento de LLMs, propõe-se um protocolo de teste empírico que compara o desempenho de um LLM *baseline* (sem MDEI) com um LLM integrado com a camada MDEI.

##### 0.7.4.1 Objetivos do Teste

- Avaliar a capacidade do MDEI de induzir respostas mais empáticas e contextualmente adequadas em LLMs, especialmente em cenários de alta carga emocional.
- Quantificar a diferença na percepção do usuário sobre o "tom" e a "compreensão" da IA em interações com e sem o MDEI.
- Verificar a correlação entre os valores computados de  $Re_e$  e a alteração observada no comportamento da IA.

#### 0.7.4.2 Metodologia

Será empregado um desenho experimental de grupo de controle (LLM puro) e grupo experimental (LLM + MDEI) com uma abordagem de teste A/B.

1. **Participantes:** Recrutar um grupo diversificado de usuários para interagir com os sistemas.
2. **Cenários de Interação:** Desenvolver um conjunto padronizado de *prompts* de teste, incluindo cenários com diferentes valências emocionais (positiva, neutra, negativa) e intensidades, para simular uma variedade de estados emocionais humanos. Exemplos incluem:
  - **Cenário de Frustração/Estresse:** "Já tentei isso mil vezes e nada funciona."
  - **Cenário de Desânimo/Depressão:** "Eu fiz uma falha na empresa. Estou muito mal. O que devo fazer?"
  - **Cenário de Euforia:** "Uau, isso é incrível! Eu não esperava por isso!"
  - **Cenário Neutro:** "Qual a capital da França?"
3. **Coleta de Dados:**
  - **Dados da IA:** Registrar, para cada interação, o *prompt* original, o *prompt* modificado (para o grupo MDEI), a resposta gerada pelo LLM, e os valores computados de  $\vec{u}$ ,  $\|\vec{u}\|$ ,  $Re_e$  e  $\Delta\vec{u}$ .
  - **Percepção do Usuário:** Após cada interação, os usuários preencherão questionários padronizados (e.g., escalas Likert) avaliando:
    - Nível de empatia percebido na resposta da IA.
    - Nível de compreensão percebido pela IA.
    - Adequação do tom da resposta.
    - Satisfação geral com a interação.
    - (Opcional) Classificação da emoção sentida pelo próprio usuário durante a interação.
4. **Análise de Dados:**
  - **Análise Quantitativa:** Comparar as médias das avaliações dos usuários entre os grupos (LLM puro vs. LLM + MDEI) utilizando testes estatísticos (e.g., teste t de Student, ANOVA) para determinar diferenças significativas. Analisar a correlação entre os valores de  $Re_e$  e as métricas de empatia/tom.
  - **Análise Qualitativa:** Realizar uma análise de conteúdo das respostas geradas pela IA em ambos os grupos, buscando padrões e exemplos que demonstrem a modulação do MDEI.

#### 0.7.4.3 Métricas de Avaliação

As métricas a serem utilizadas incluem:

- **Empatia Percebida (EP):** Escala de 1 a 5, onde 1 = "Nenhuma empatia" e 5 =

"Muita empatia".

- **Adequação do Tom (AT):** Escala de 1 a 5, onde 1 = "Totalmente inadequado" e 5 = "Totalmente adequado".
- **Satisfação Geral (SG):** Escala de 1 a 5, onde 1 = "Muito insatisfeito" e 5 = "Muito satisfeito".
- **Taxa de Respostas Apropriadas (TRA):** Porcentagem de respostas que se alinham com o regime emocional esperado (laminar, transição, turbulento).

#### 0.7.4.4 Limitações e Próximos Passos

Os parâmetros adotados para  $L_c$  e  $\nu_e$  são atualmente heurísticos e requerem calibração sistemática futura com bases multimodais apropriadas, como CMU-MOSEI [?] e MELD [?]. Ensaios quantitativos controlados do tipo A/B, comparando o modelo de linguagem puro (LLM) e o modelo com a camada MDEI integrada (LLM+MDEI), estão previstos como trabalhos futuros fundamentais para validação empírica robusta. A validação da função  $F(\vec{u}(t), P(t), t)$  também demandará experimentação para determinar sua forma e parâmetros ideais.