Control and identification of a fourth order fluid level system with neural networks and reinforcement learning

Lucas Guilhem de Matos

Orientador: Prof. Dr. Adolfo Bauchspiess

Sumário

- Motivação
- Sistema escolhido para experimentos
- Proposta do trabalho: Controlador com Aprendizado por Reforço e Redes Neurais
  - Estimação de funcão com redes neurais - Preditor
  - Elementos dos aprendizado por reforço no controle.
  - Método Ator-Crítico
  - Estrutura do controlador
- Resultados experimentais
- Conclusão

Motivação – Controle Otimo

- Objetivo Ideal.
- Critério de Otimalidade preestabelecido.
- Resolução das equações de Hamilton-Jacobi-Bellman. Muito difícil ou, em alguns casos, impossível.
- Necessita conhecimento completo da dinâmica do sistema.
- Funciona apenas para uma dinâmica específica.

Motivação – Controle Adaptativo

- Consegue lidar com variação de parâmetros ou com dinâmica desconhecida.
- As leis de controle mudam ao longo do tempo.
- Identificação de parâmetros.

Motivação - Aprendizado por Reforço

- Algorítmo de aprendizado adaptativo.
- Raízes na teoria de controle ótimo.
  - Sutton & Barto 1992
  - Vrabie, Vamvoudakis & Lewis 2013
- Simples implementação.

Aprendizado por Reforço - Conceito

- Psicologia comportamental.
- Tentativa e erro.
- Recompensa e punição.
- Aprendizado não supervisionado.

Aprendizado por Reforço - Elementos

- Estados: A priori a formulação do problema é feita de forma discreta.
- Agente: O elemento que executa ações e obtém a experiência.
- Política de Ações: Regras que o agente usa para determinar uma ação a partir de um estado. Procura e memória. (Exploração X Exploração).
- Ambiente: O elemento que define o estado e dá a recompensa.
- Recompensa: Resultado imediato de uma ação.
- Valor: Espectativa de recompensa a ser atingida a partir de um determinado estado.
- Função de valor: A Função que mapeia os valores de cada estado.

Diferença Temporal

$$V(s_k) = \gamma r_{k+1} + \gamma^2 r_{k+2} + \ldots + \gamma^\infty r_{k+\infty}$$

$$V(s_k) = \sum_{i=0}^{\infty} \gamma^i r_{k+i+1}$$

$$V(s_k) = r_{k+1} + \sum_{i=1}^{\infty} \gamma^i r_{k+i+1}$$

$$V(s_k) = r_{k+1} + \gamma \sum_{i=1}^{\infty} \gamma^i r_{k+i+2}$$

$$V(s_k) = r_{k+1} + \gamma V(s_{k+1})$$

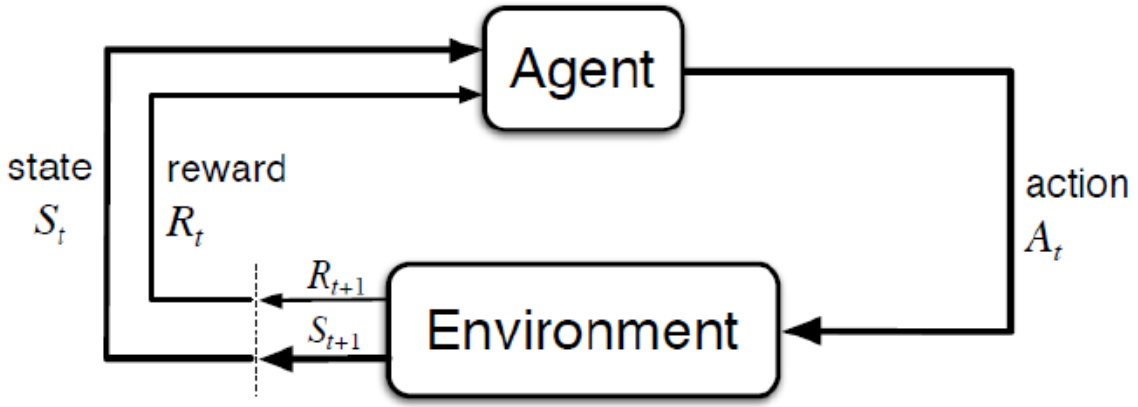
Diferença Temporal

$$V(s_t) = r_{t+1} + \gamma V(s_{t+1})$$

which states that the sum of the discounted estimated value of the next state value and the next state reward is the target to the actual estimated value. The error between the target and the actual estimated value $\bar{V}$ is called *Temporal Difference Error,* denoted by δ_{td}, that is;

$$\delta_{td} = r_{t+1} + \gamma \bar{V}(s_{t+1}) - \bar{V}(s_t).$$

The δ_{td} is the value that will be used as parameter for updating the value function and/or the action policy. In this work, it is the value used to update the weights of the neural networks.

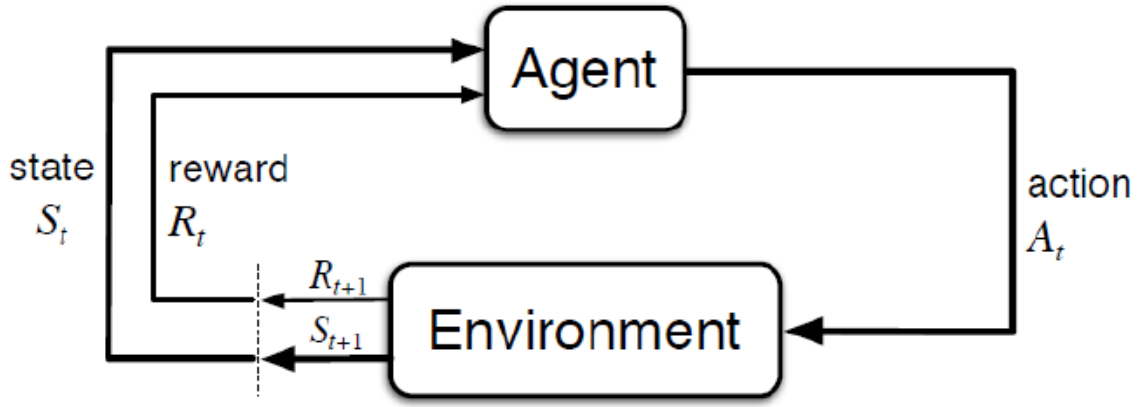


SARSA - State-Action-Reward-State-Action

$$\delta_{td} = r_{t+1} + \gamma \bar{V}(s_{t+1}) - \bar{V}(s_t).$$

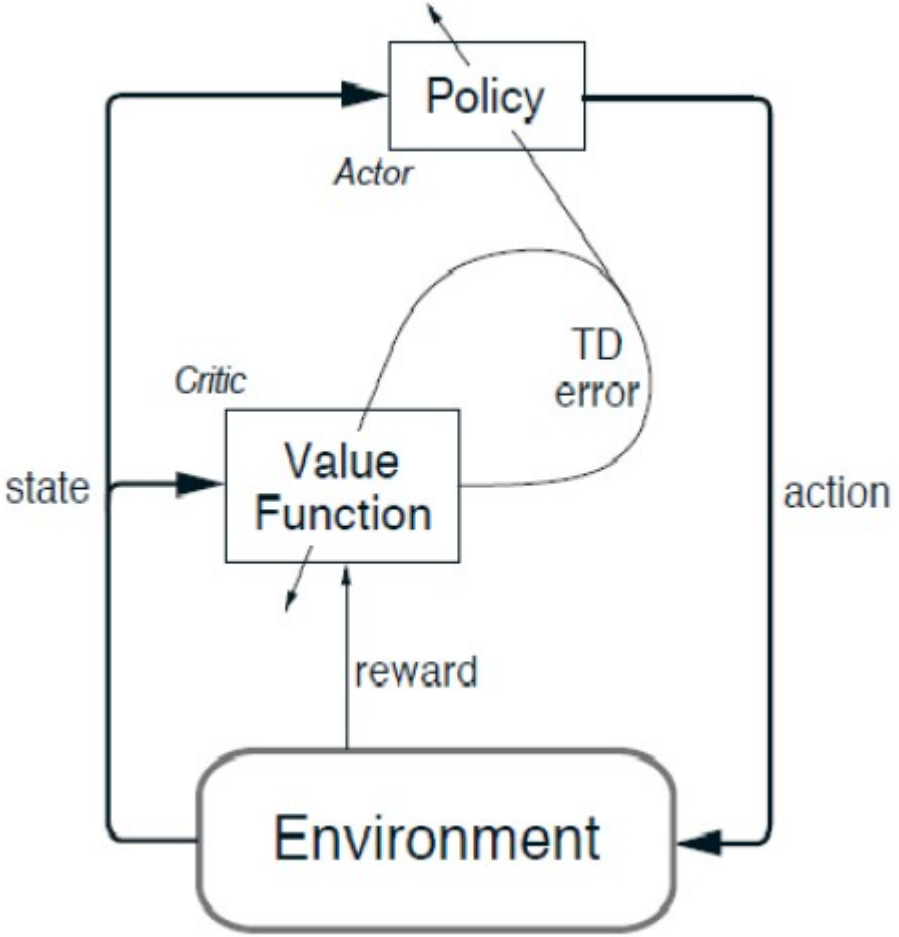
As seen in Sutton and Barto (1998), the action will be selected by the action policy and only then the value of taking that action in this specific state will be estimated. Also, in order to estimate the next value so the target for update can be complete, the action that would be taken in the consecutive state needs to be defined using the same policy. The value of a state-action pair is denoted as $Q(s_t, a_t)$. In this case δ_{td} is given by:

$$\delta_{td} = r_{t+1} + \gamma \bar{Q}(s_{t+1}, a_{t+1}) - \bar{Q}(s_t, a_t)$$



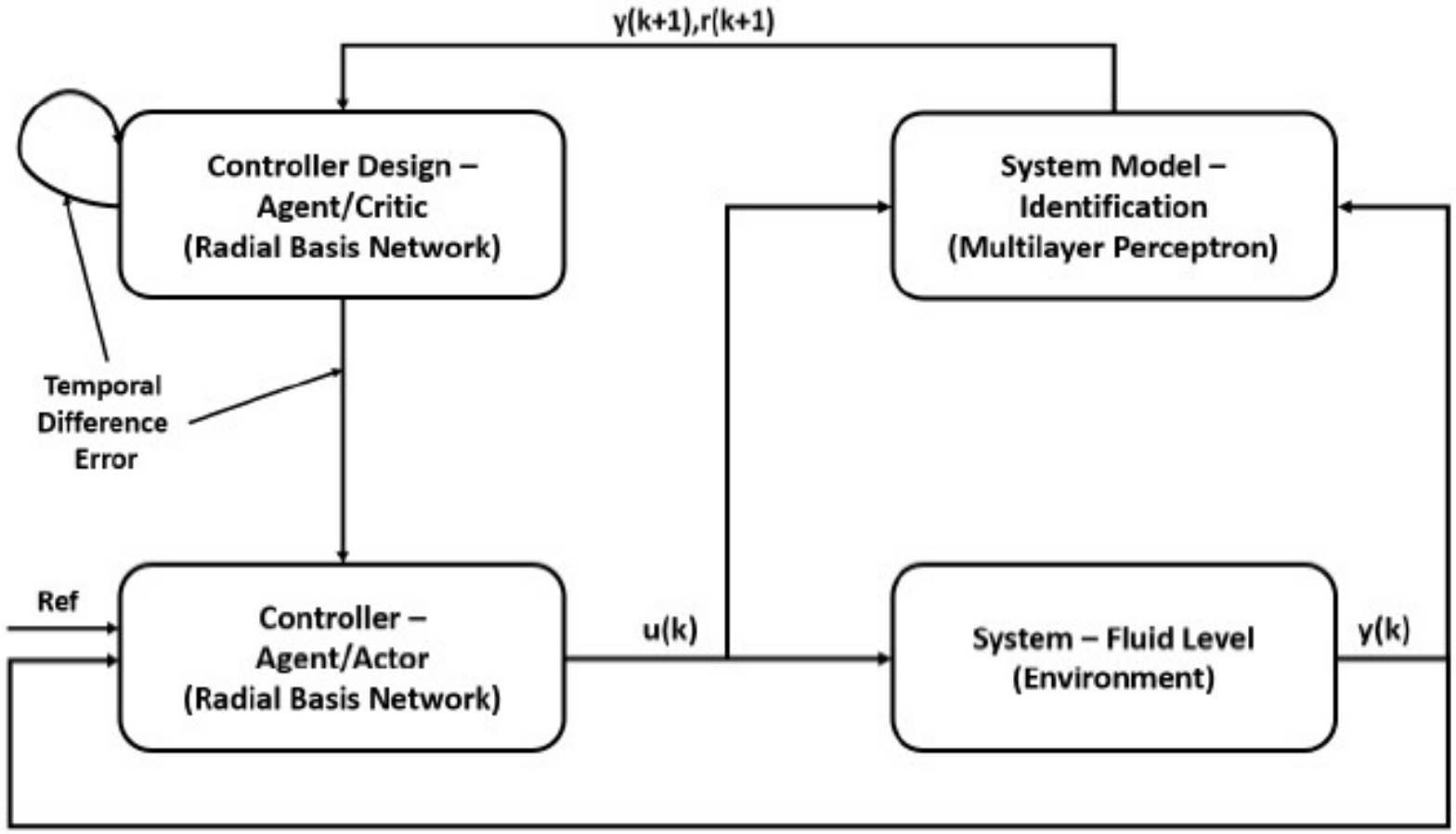
Actor-Critic Method

The actor-critic is a TD method that has s**eparated structures for the action policy**, known as Actor, and the **value function estimation**, known as Critic. The Actor is the structure that defines the actions that will be taken and the critic is the structure responsible for evaluating the Actor's decisions. Notice that both the actor and the critic are part of the agent as none of them are part of the environment. Both elements learn to achieve the same goal, but represent different concepts.

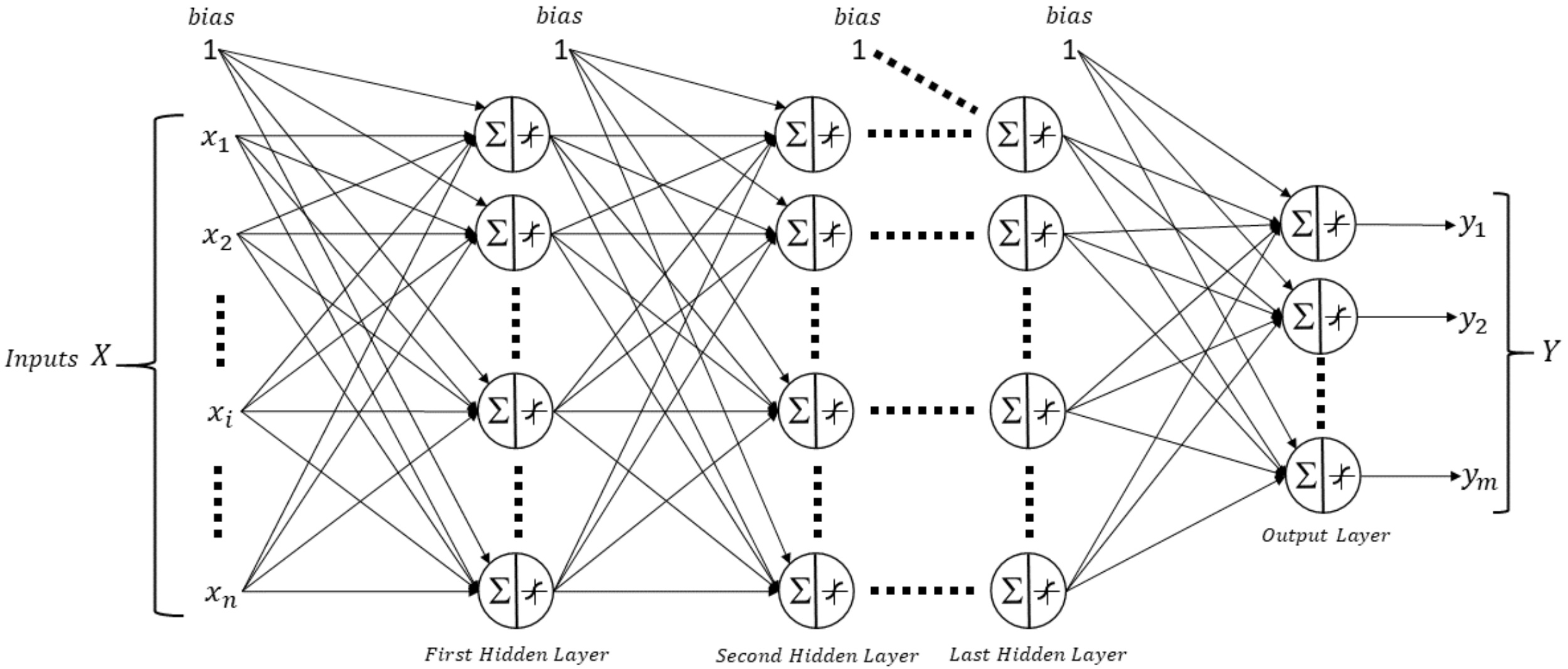


Actor-Critic method as seen in Sutton and Barto (1998).

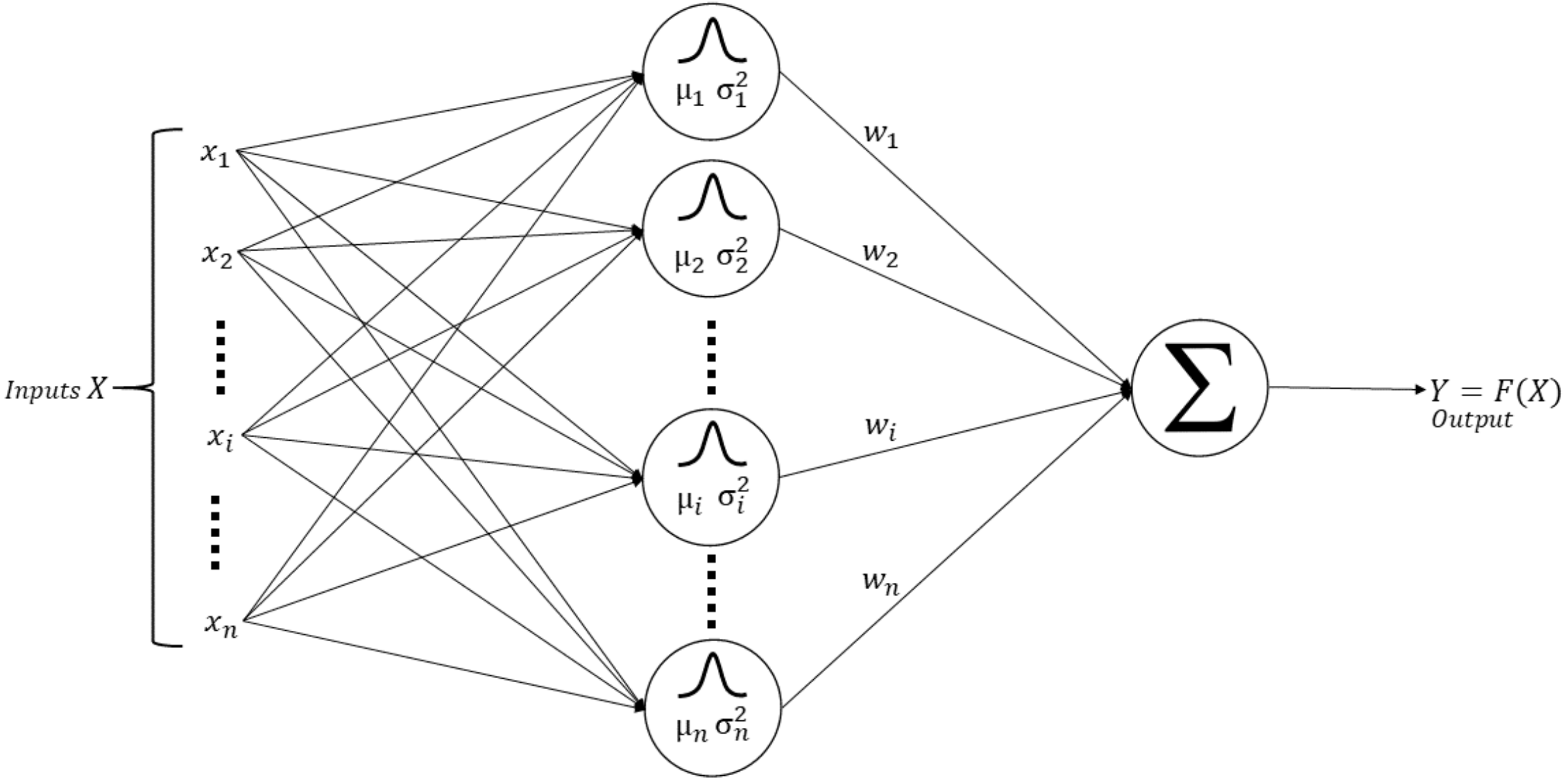
Actor-Critic RL-Controller



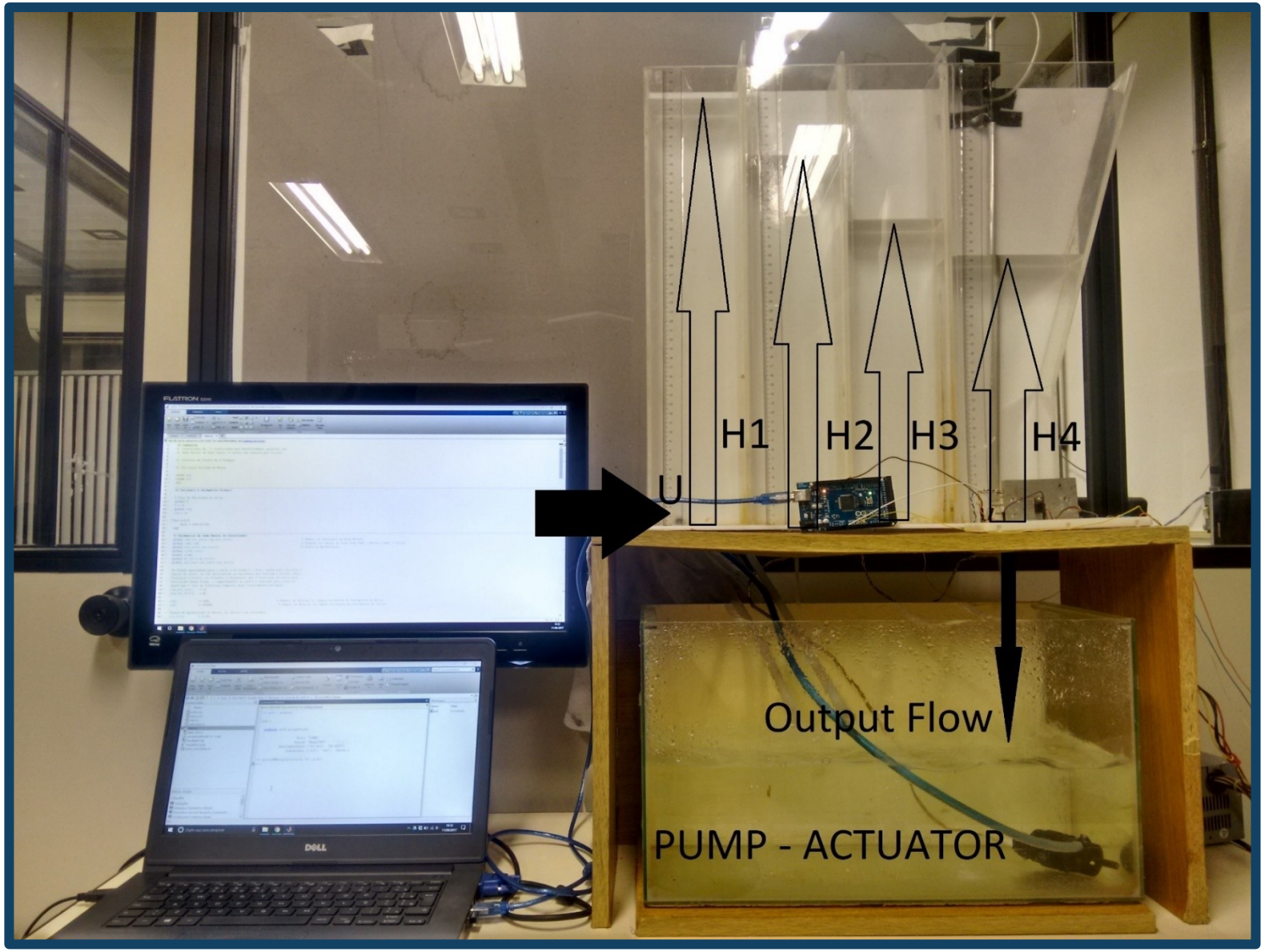
Redes Neurais – Perceptron Multicamada



Redes Neurais – Base Radial



Processo Experimental



Base dos tanques:

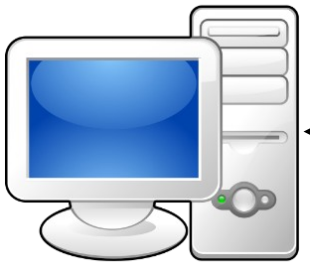
Largura: 10 cm

Comprimento: 6 cm

Altura dos Tanques: 50 cm

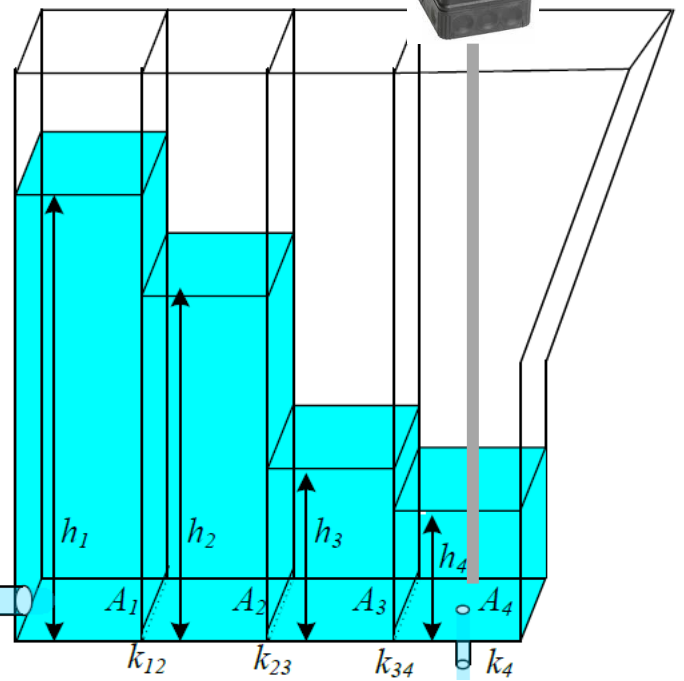
Montagem Completa

Arduino – Interface

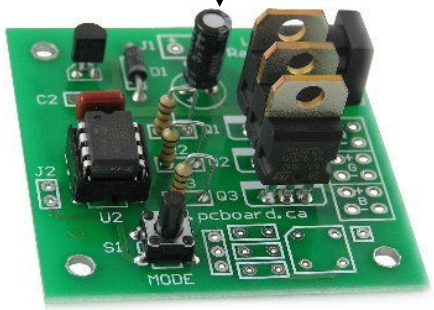


MatLab®

Sensor



Bomba – Atuador

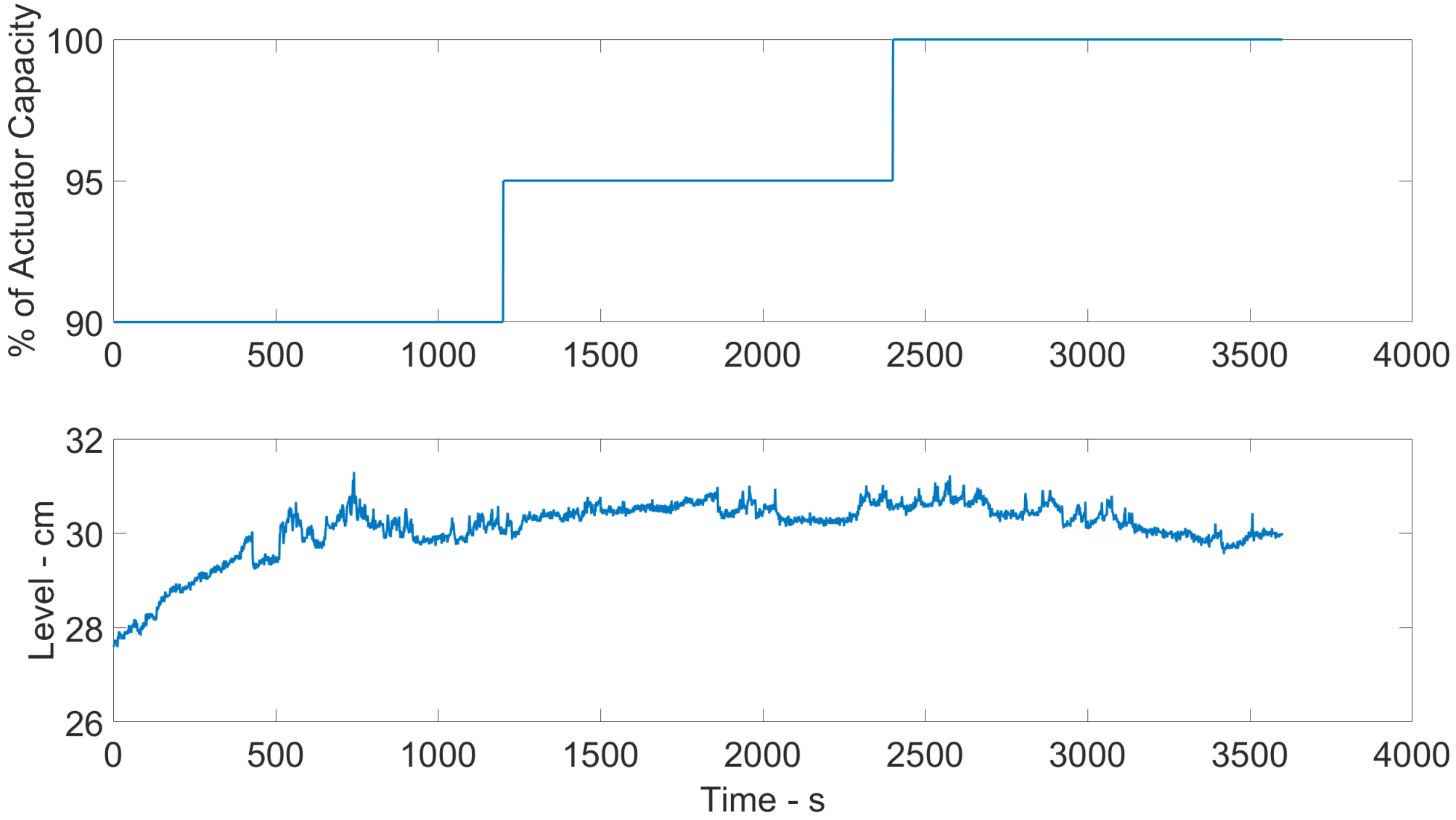


Circuito PWM

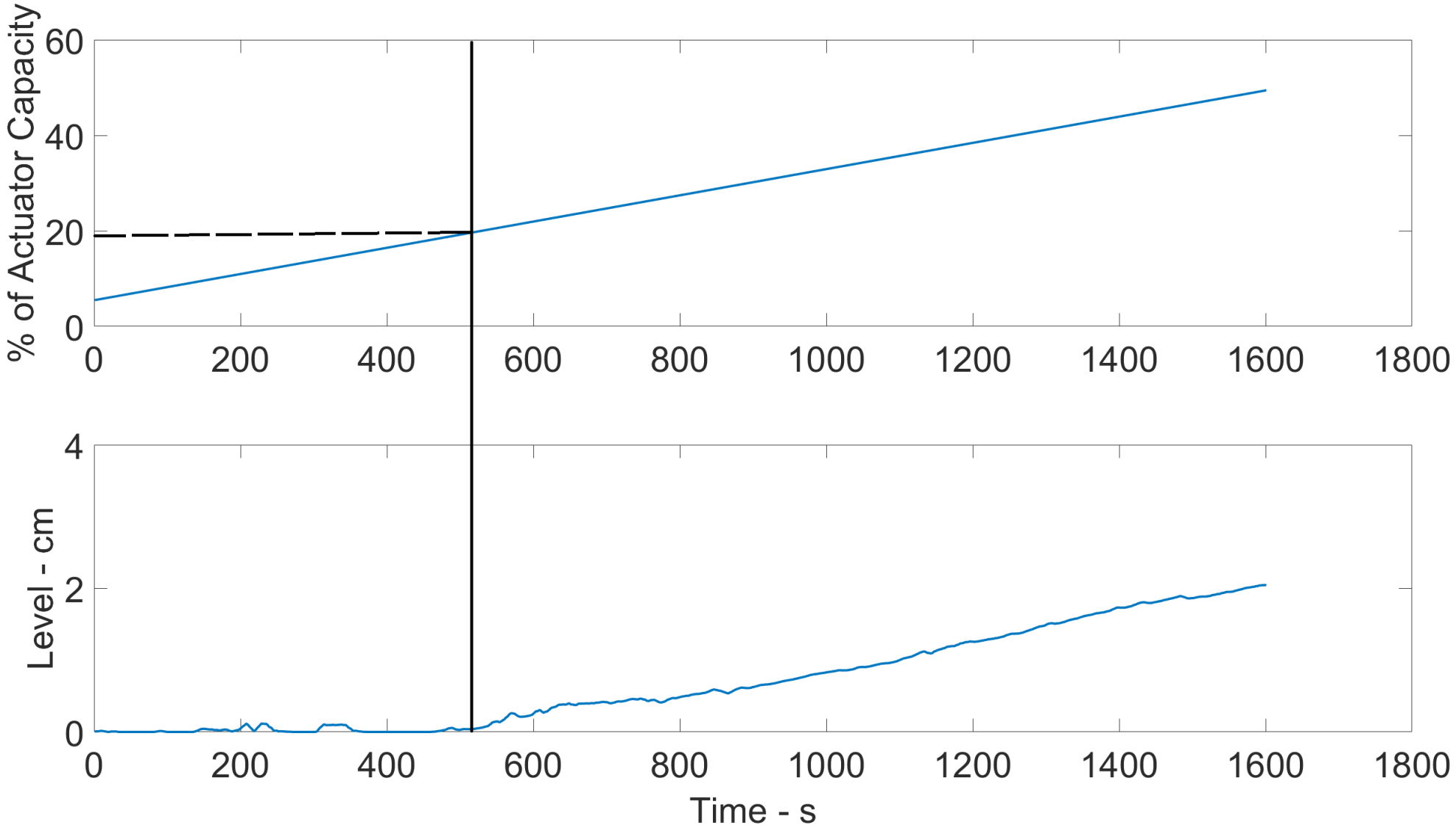
Motivos para a Escolha do Sistema de 4 Tanques

- Grande variação das constantes de tempo (“stiff” differential eq.).
- Não Linearidades como saturação, zona morta e atraso consideráveis.
- Possibilidade de variação dos parâmetros com válvulas reguláveis.
- Processo típico em diversos ramos industriais:
  - Tratamento de água
  - Químico
  - Petroquímico
  - ...

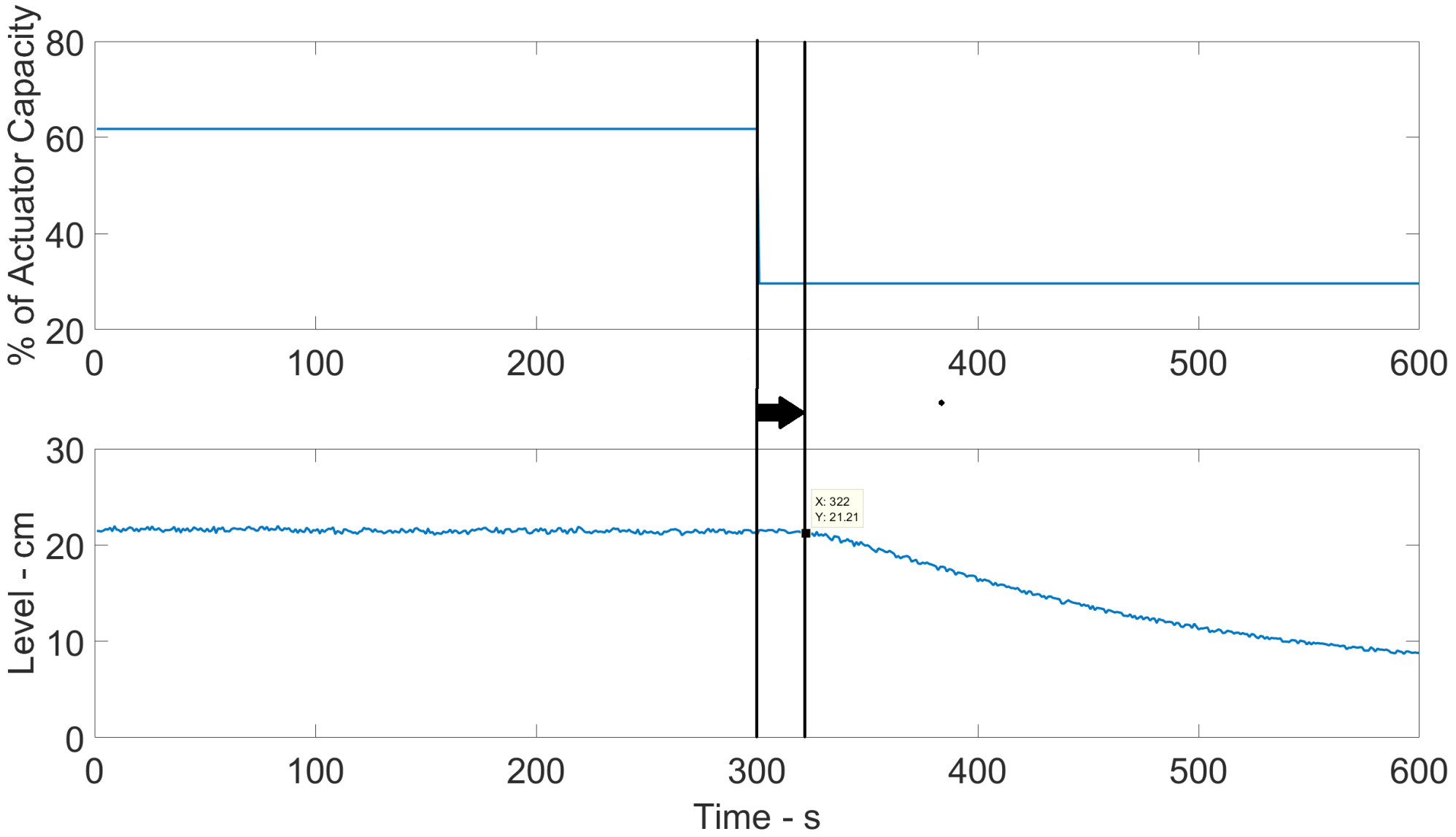
Saturação


% of Actuator Capacity
100
95
90
0
500
1000
1500
2000
2500
3000
3500
4000
Level - cm
32
30
28
26
0
500
1000
1500
2000
2500
3000
3500
4000
Time - s

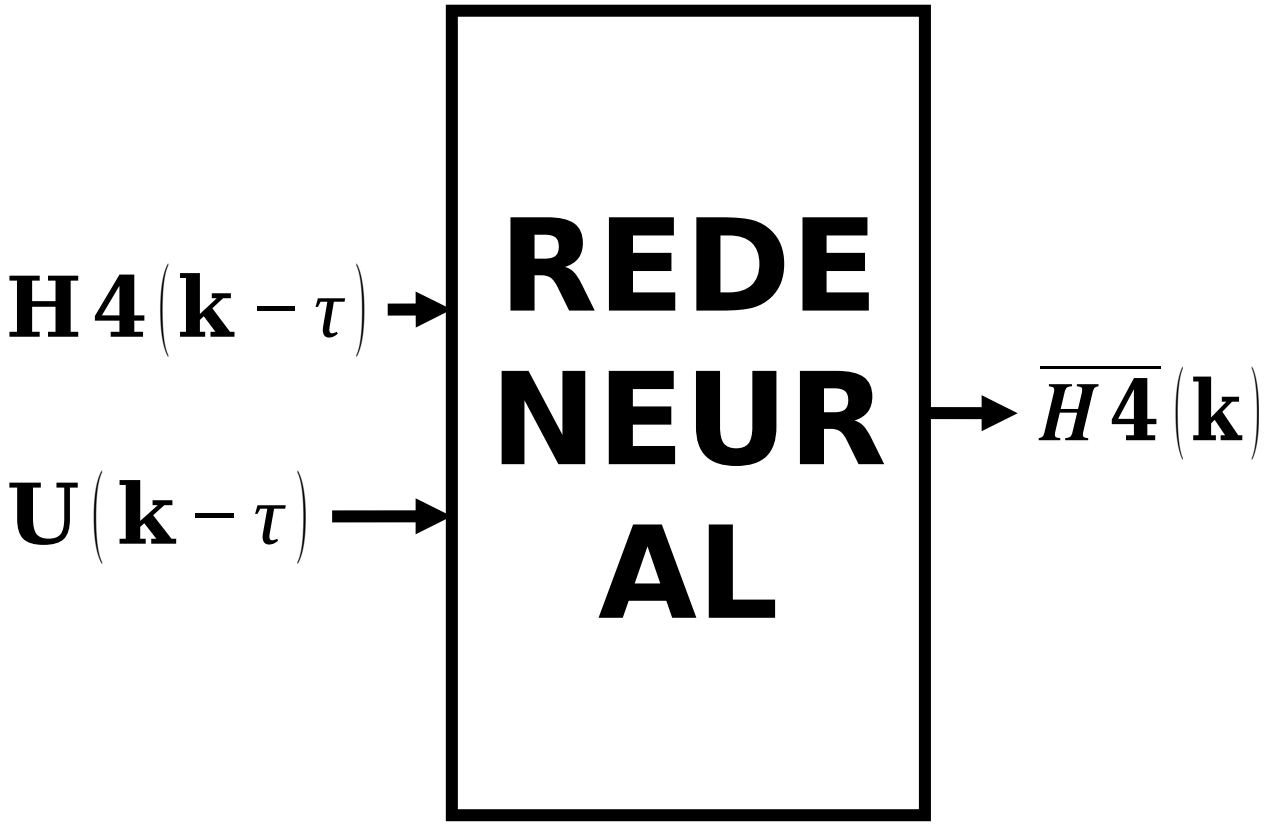
Zona Morta


% of Actuator Capacity
60
40
20
0
0
200
400
600
800
1000
1200
1400
1600
1800
Level - cm
4
2
0
0
200
400
600
800
1000
1200
1400
1600
1800
Time - s

Atraso

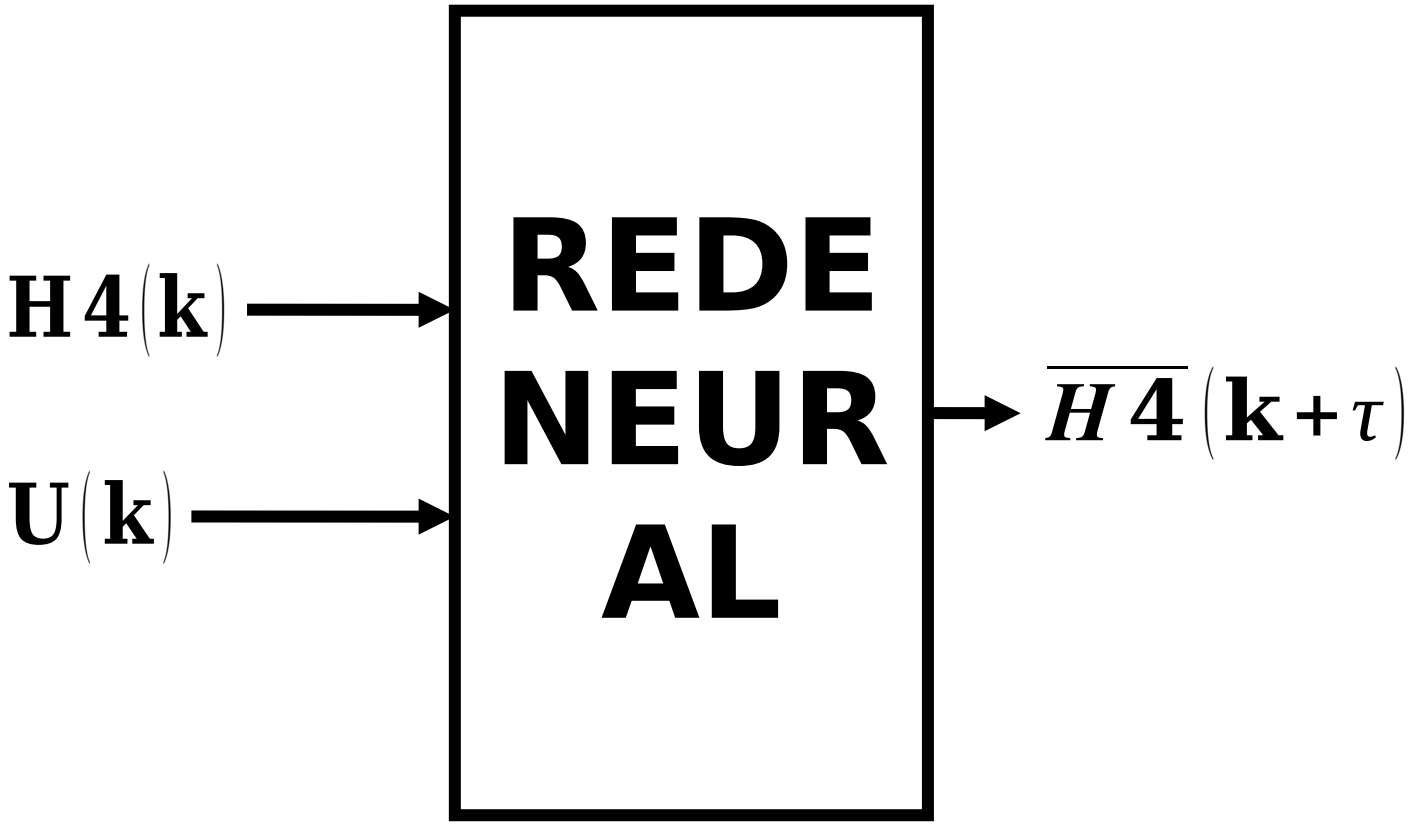

% of Actuator Capacity
80
60
40
20
0
100
200
400
500
600
Level - cm
30
20
10
0
X: 322
Y: 21.21
300
Time - s

Identificação do Sistema Para Predição



- Premissa do Trabalho: O atraso médio do sistema é conhecido (ensaios prévios). .
- O objetivo é encontrar o modelo dinâmico, para estimar a saída futura.

Identificação do Sistema Para Predição



- Premissa do Trabalho: O atraso médio do sistema é conhecido (ensaios prévios). .
- O objetivo é encontrar o modelo dinâmico, para estimar a saída futura.

Controle via Aprendizado por Reforço

- O agente é o próprio controlador, a política de ações, os ganhos de um controlador PI por exemplo, ou, no caso deste trabalho, a(s) rede(s) neural(is) que definem o controlador.
- O ambiente é o processo, o sensor, a bomba e o preditor*.
- A função de recompensa é definida a partir do módulo do erro. 4 Funções diferentes foram utilizadas.
- SARSA para calcular o valor do estado.

S.A.R.S.A.

- Ao invés de calcular o valor apenas do estado, calcula-se o valor de um par estado-ação, significando o valor de se escolher uma determinada ação em um determinado estado.

$$\delta_{td} = r_{t+1} + \gamma \bar{V}(s_{t+1}) - \bar{V}(s_t).$$

Se torna

$$\delta_{td} = r_{t+1} + \gamma \bar{Q}(s_{t+1}, a_{t+1}) - \bar{Q}(s_t, a_t)$$

Funções de Recompensa

RF1

*r*k = - alfa if e(k) > e(k-1)
*r*k = 0 if e(k) < e(k-1)

RF2

*r*k = - alfa if e(k) – e(k-1) < epsilon1
*r*k = -alfa-beta if e(k) – e(k-1) < epsilon2

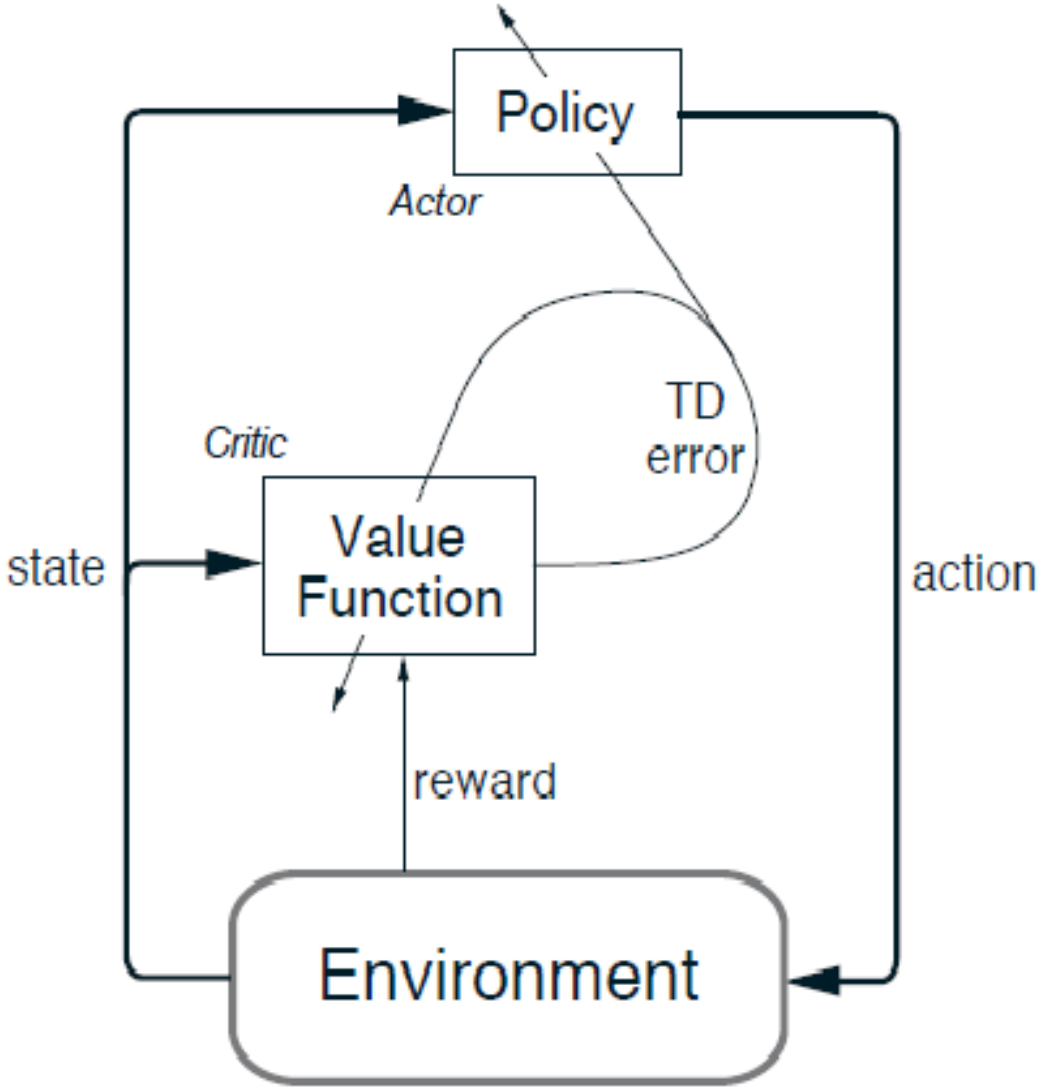
RF3

$$r_k = -|e(k)|$$

RF4

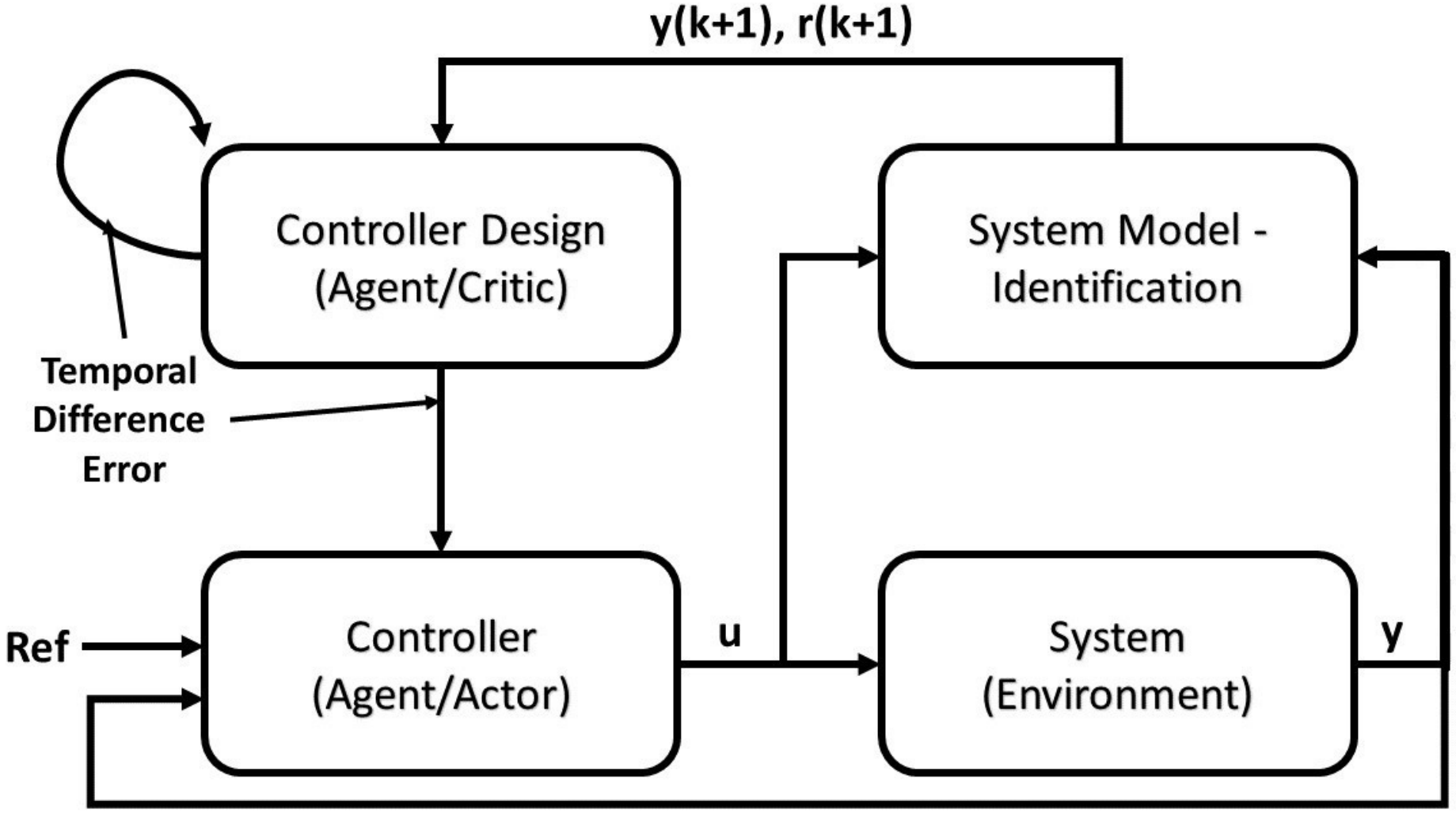
$$r_k = -2|e(k)| + |e(k-1)|$$

O Método Ator-Crítico



- Dois elementos: o Ator e o Crítico.
- Ambos são parte do Agente.
- O ator executa as ações ao passo que o crítico avalia o quão boas elas foram para alcançar o objetivo.
- No critico se calcula o erro de diferença temporal que é usado para atualizar as duas estruturas.
- Uma rede neural para cada elemento

Estrutura do Controlador



Algoritmo do Controlador

1. Inicializar e os parâmetros constantes a serem usados pelo controlador.
2. Inicializar usando a rede neural definida para o ator.
3. Inicializar usando a rede neural definida para o crítico.
4. Execute a ação no modelo para prever o próximo estado .
5. Observe a recompensa .
6. Calcule usando a rede neural definida para o ator.
7. Calcule usando a rede neural definida para o crítico.
8. Calculate .
9. Atualize as redes neurais.
10. Opcional – Atualize o modelo do sistema. (O modelo pode ser usado apenas no treinamento em batelada)
11. Execute uma nova medição no sistema real, defina um novo e calcule uma nova .
12. Repita os passos de 4 a 12.

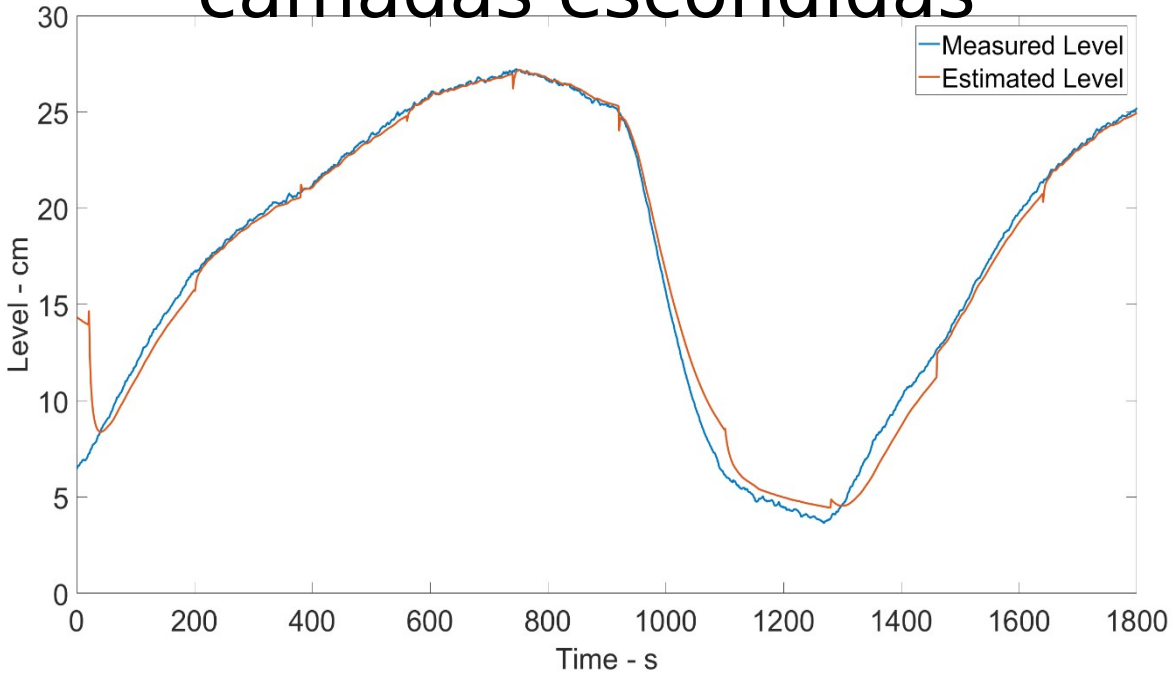
Resultados do Processo de Identificação

Network Layout/Learning Algorithm	Number of Neuron in Hidden Layers	Number of Epochs to Achieve Max Fit	Max Fit
MLP with 2 Hidden Layers	90-15	37	92.55%
MLP with 1 Hidden Layer	5	100	89.67%
RBN with Fixed Centers Started Randomly	30	50	93.05%
RBN with Fixed Centers Started Uniformly	16	43	96.82%
RBN With Randomly Started Centers Using Backpropagation	20	26	93.53%
RBN With Uniformly Started Centers Using Backpropagation	9	18	94.35%
RBN With Randomly Started Centers Using Clusters	30	22	93.88%
RBN With Uniformly Started Centers Using Clusters	36	50	95.51%
RBN with Online Creating Centers but no Center Correction	15	35	67%
RBN with Online Creating Centers with Center Correction	15	43	93.28%

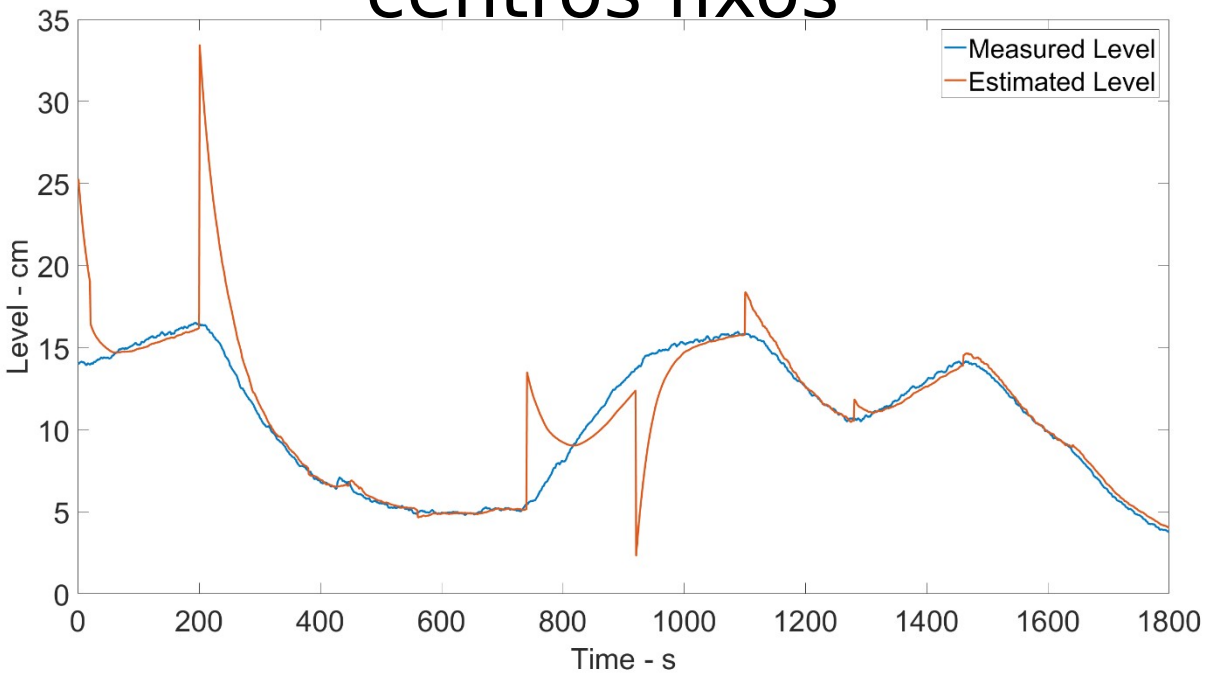
- O Número de neuronios em cada camada dos perceptrons multicamada e nas redes de base radial variou de 5 a 100.
- Apenas as melhores estruturas de rede para cada um dos casos foi escolhida para comparação.
- As redes foram avaliadas em questao de estabilidade e velocidade de resposta à falhas.

Resultados do Processo de Identificação

Perceptron Multicamada com 2 camadas escondidas



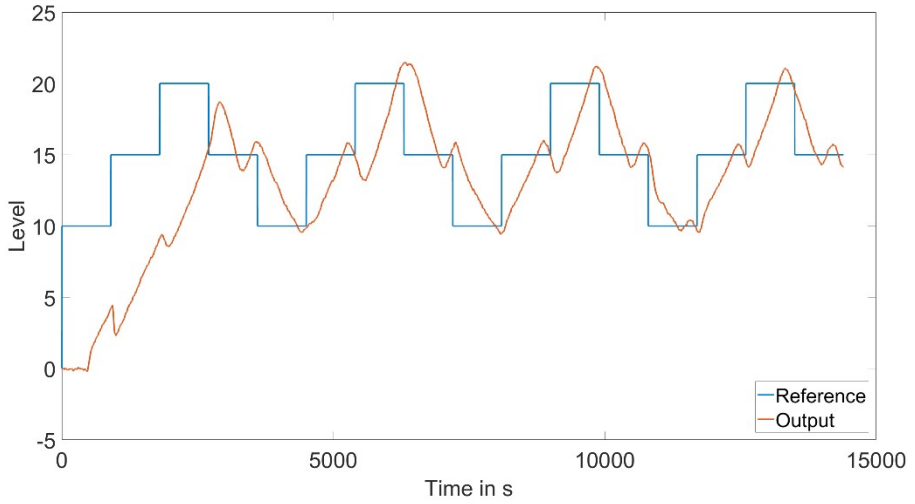
Rede de Base Radial com centros fixos



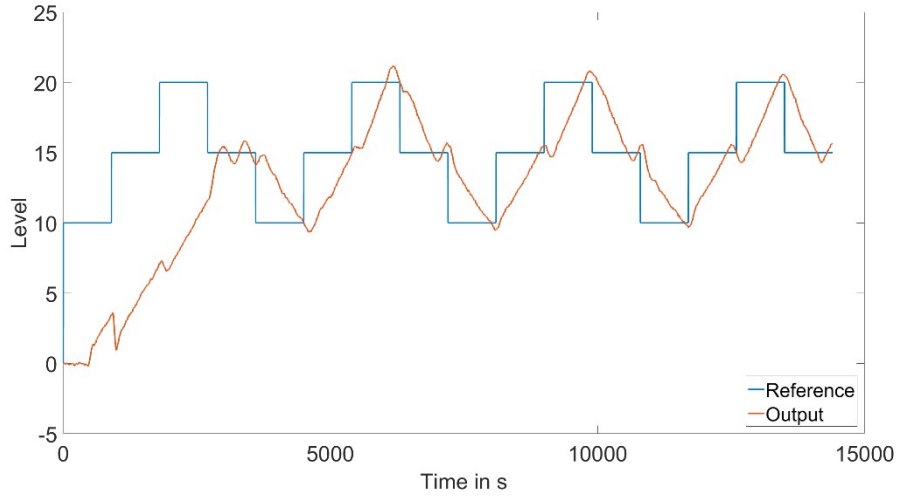
Resultados de Controle

- Período de Amostragem padrão a priori: 1s.
- Taxa de Aprendizado padrão a priori: 0.01.
- Condições iniciais (pesos e centros das redes neurais) incialmente sempre as mesmas.

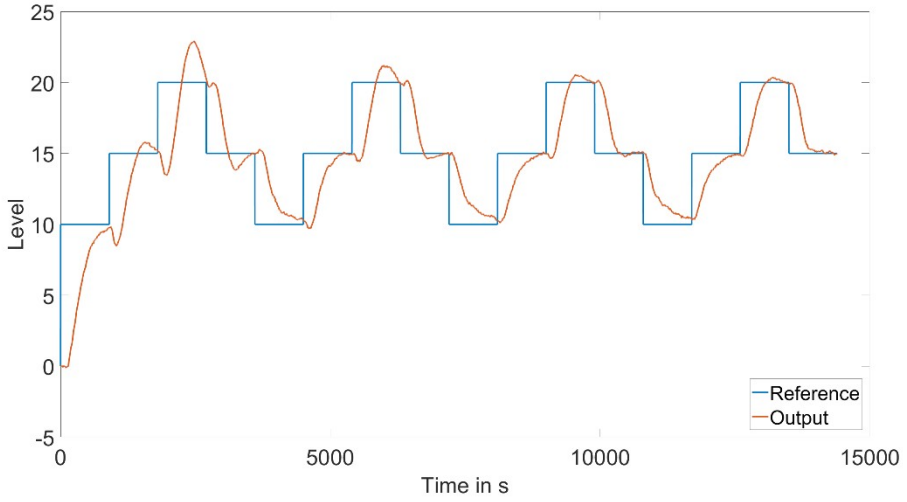
Diferentes Funções de Recompensa



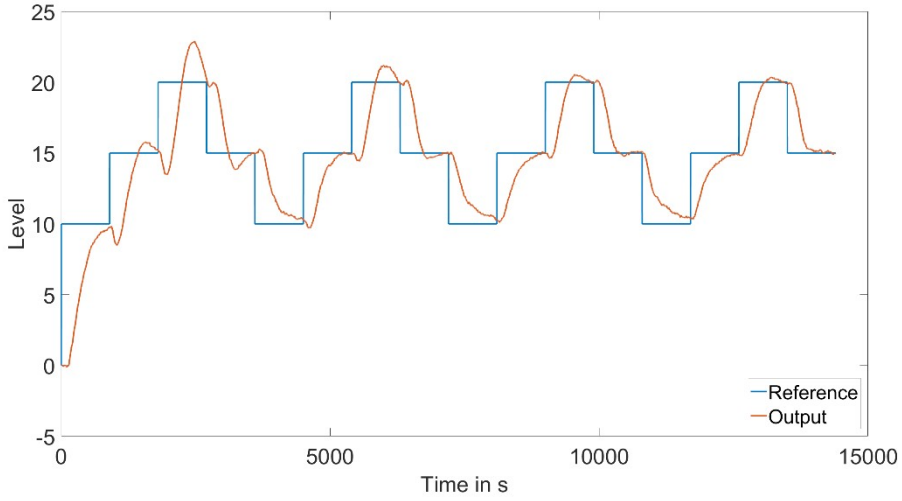
RF1



RF2

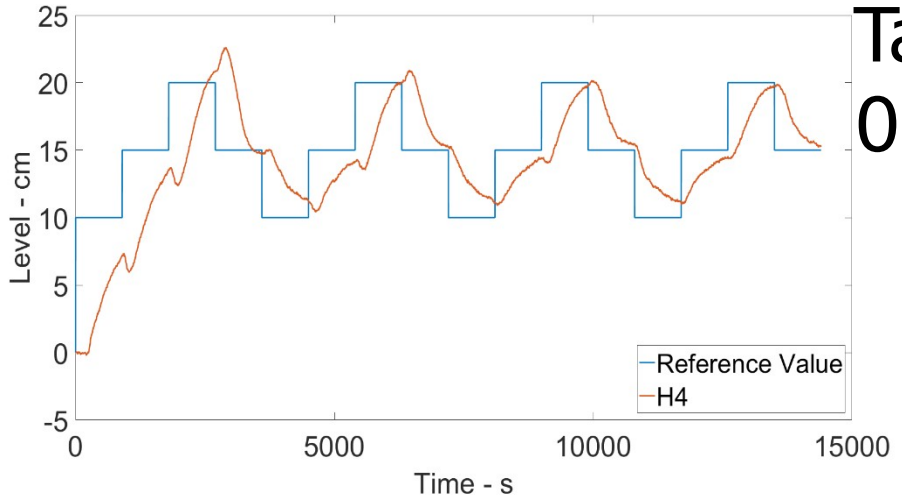


RF3

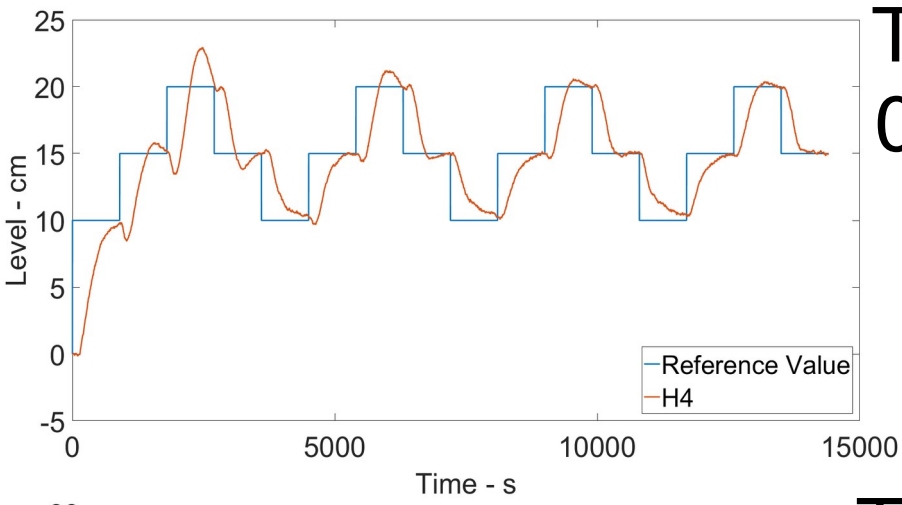


RF4

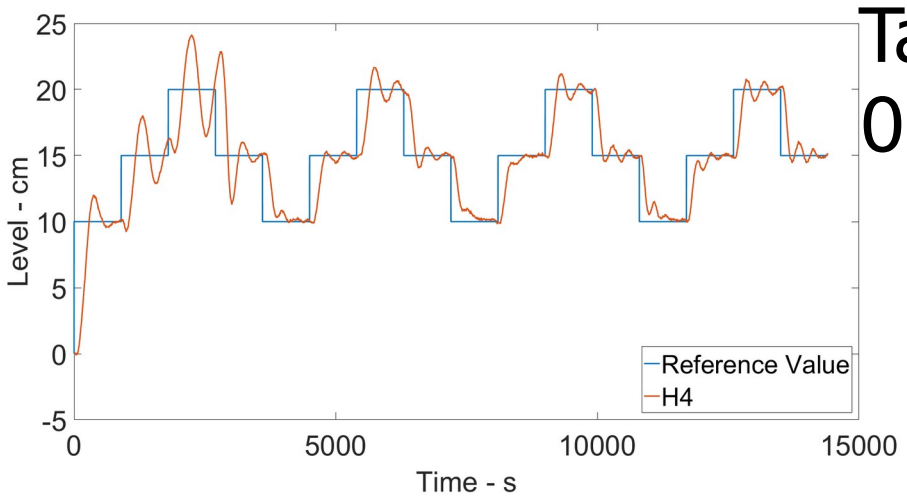
Várias Taxas de Aprendizado para RF3



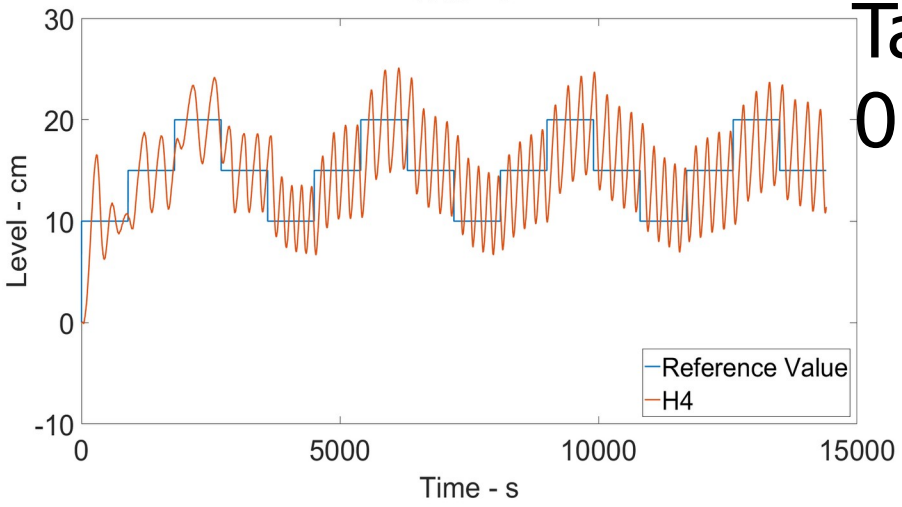
Tap
0.005



Tap
0.01



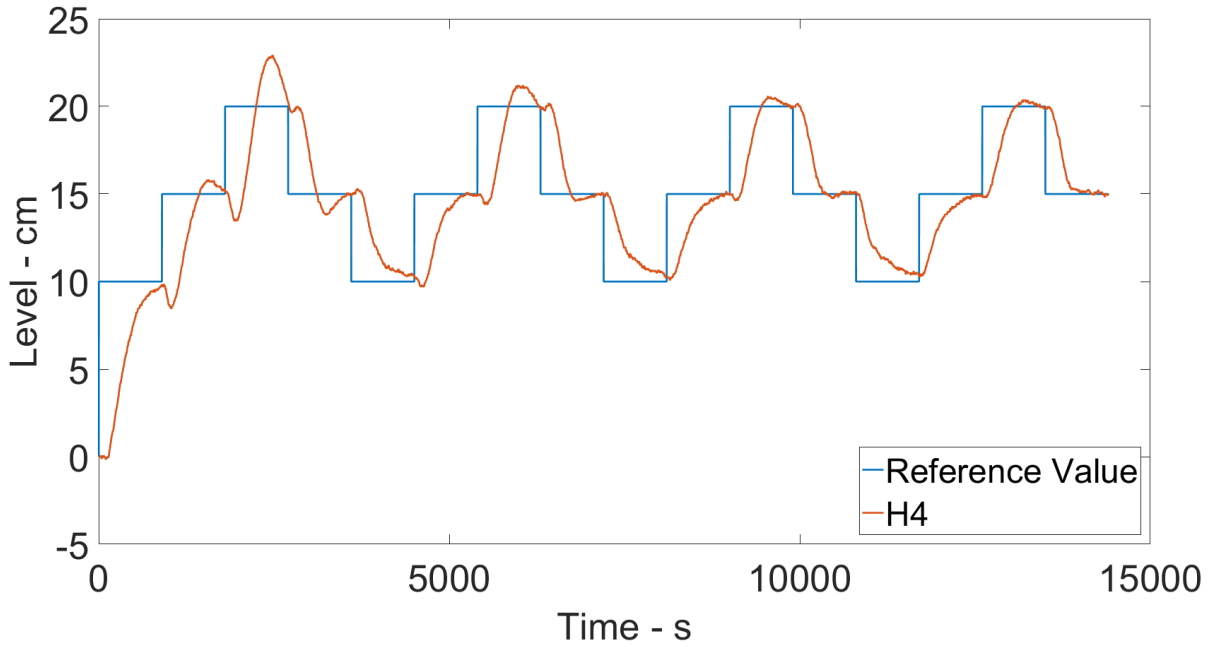
Tap
0.025



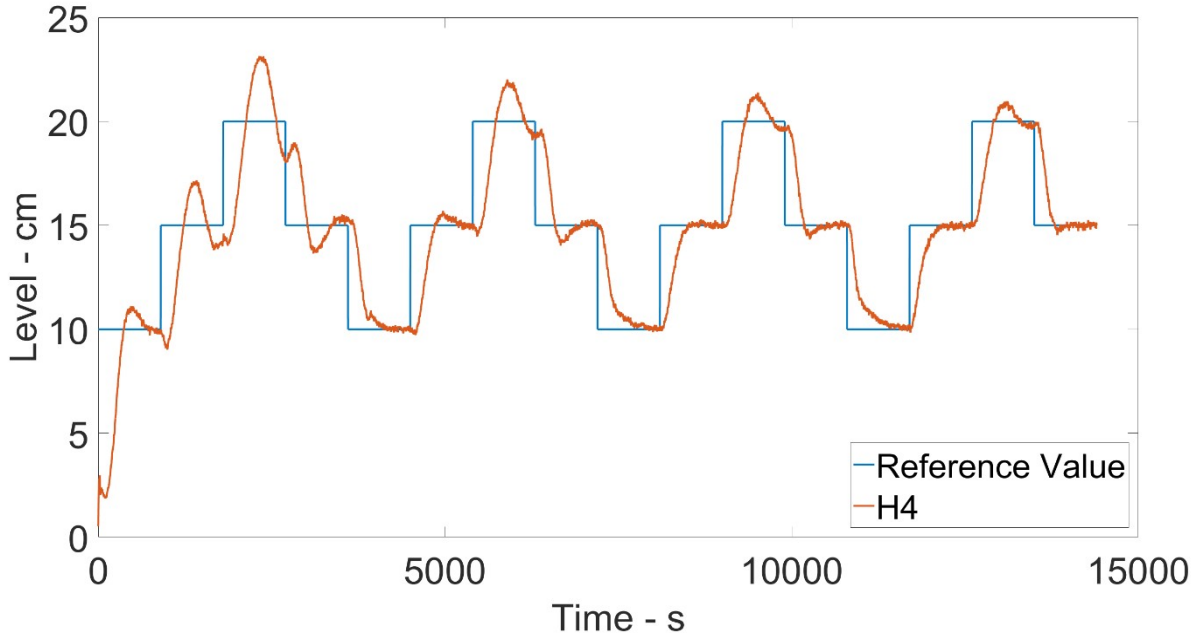
Tap
0.05

Validação das Simulações em Ambiente Real

Simulação RF4 com
Tap = 0.01 e Ts = 1s

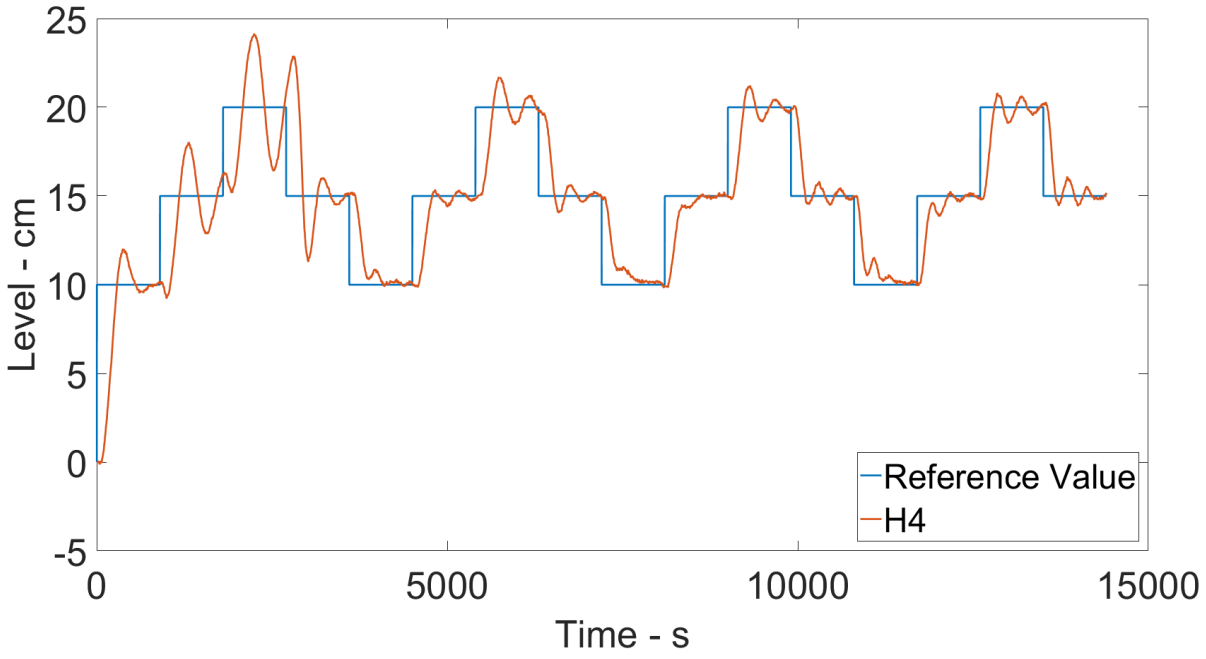


Real RF4 com
Tap = 0.01 e Ts = 1s

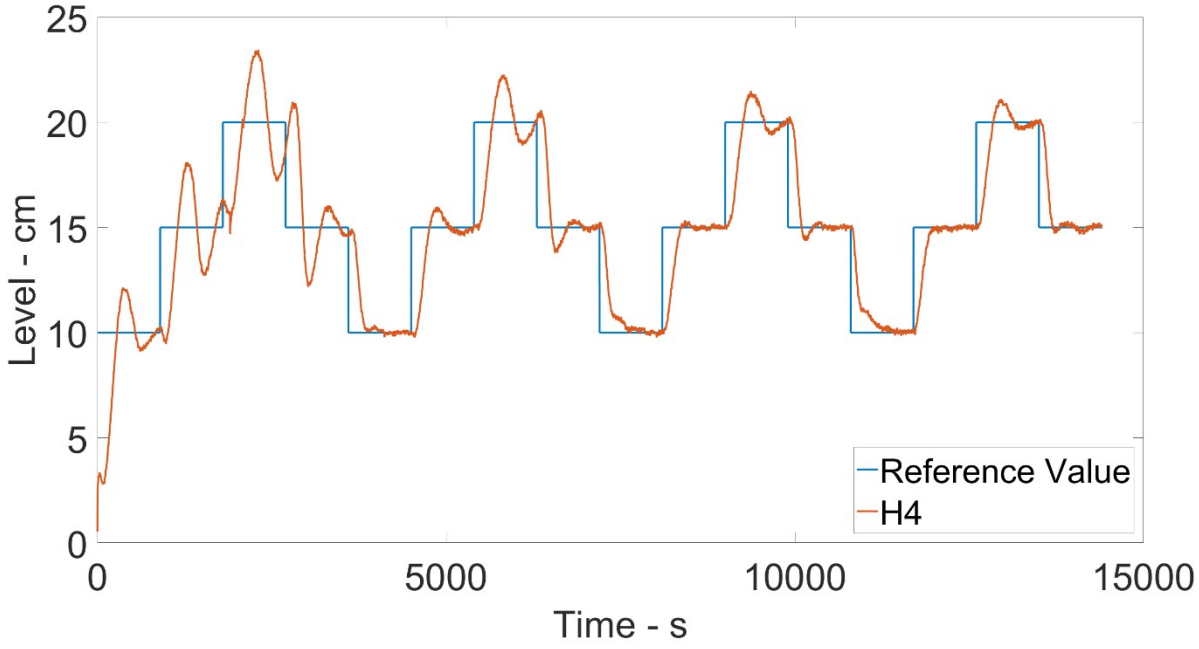


Validação das Simulações em Ambiente Real

Simulação RF3 com
Tap = 0.025 e Ts = 1s

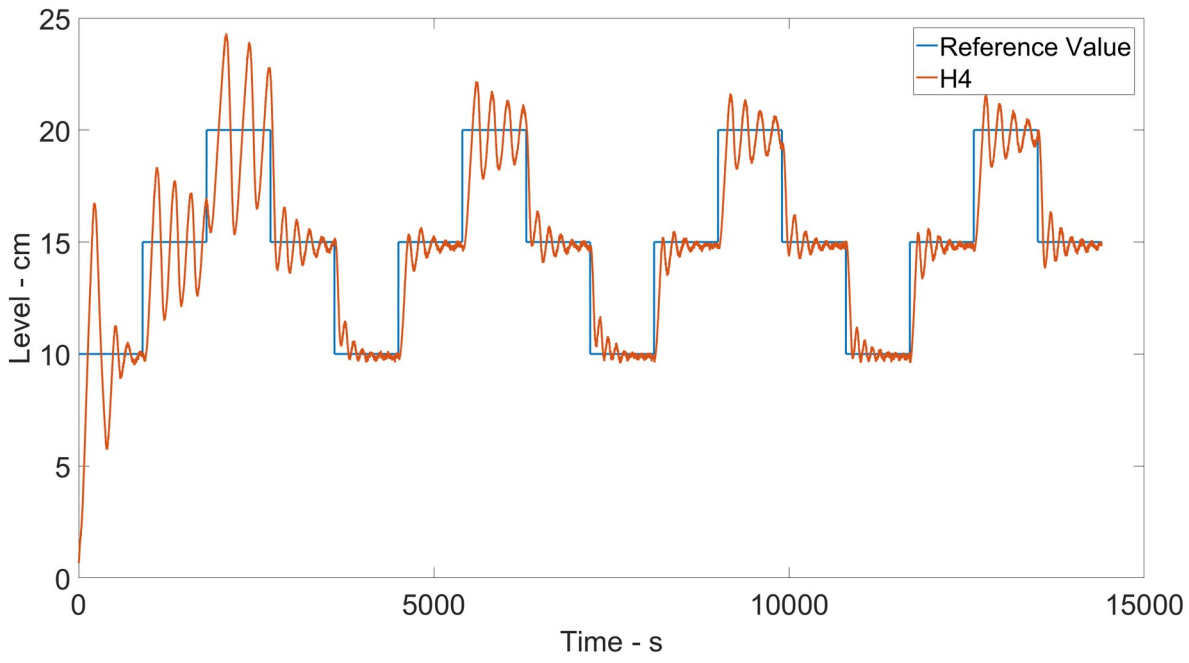


Real RF3 com
Tap = 0.025 e Ts = 1s

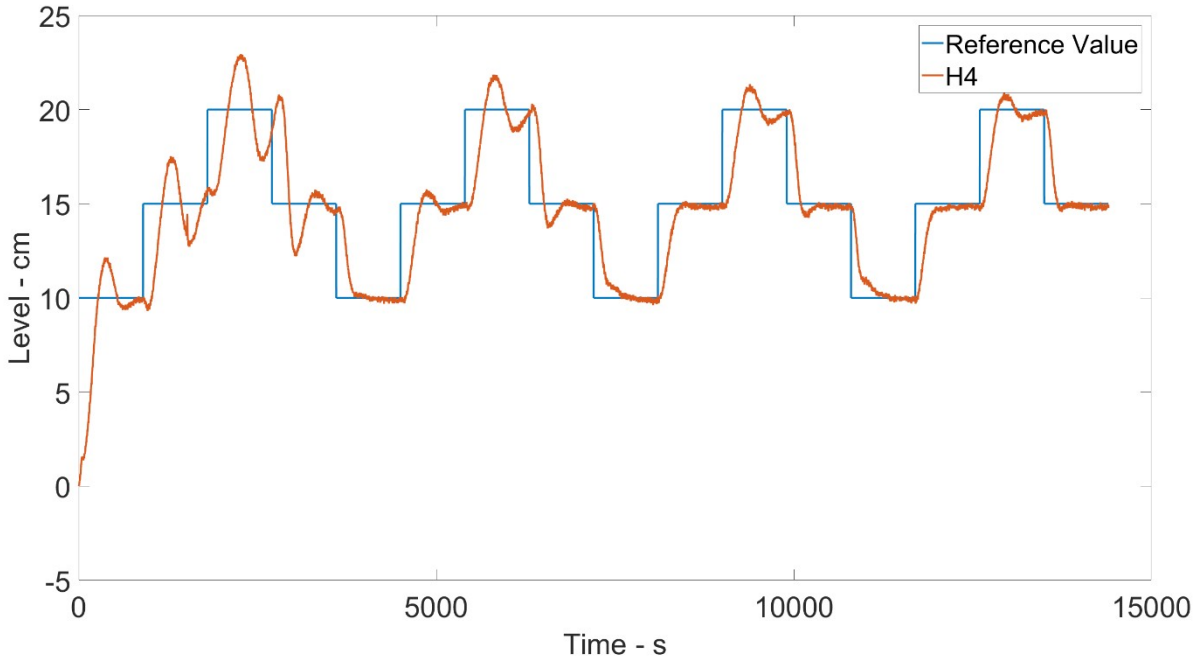


Efeitos da Taxa de Amostragem

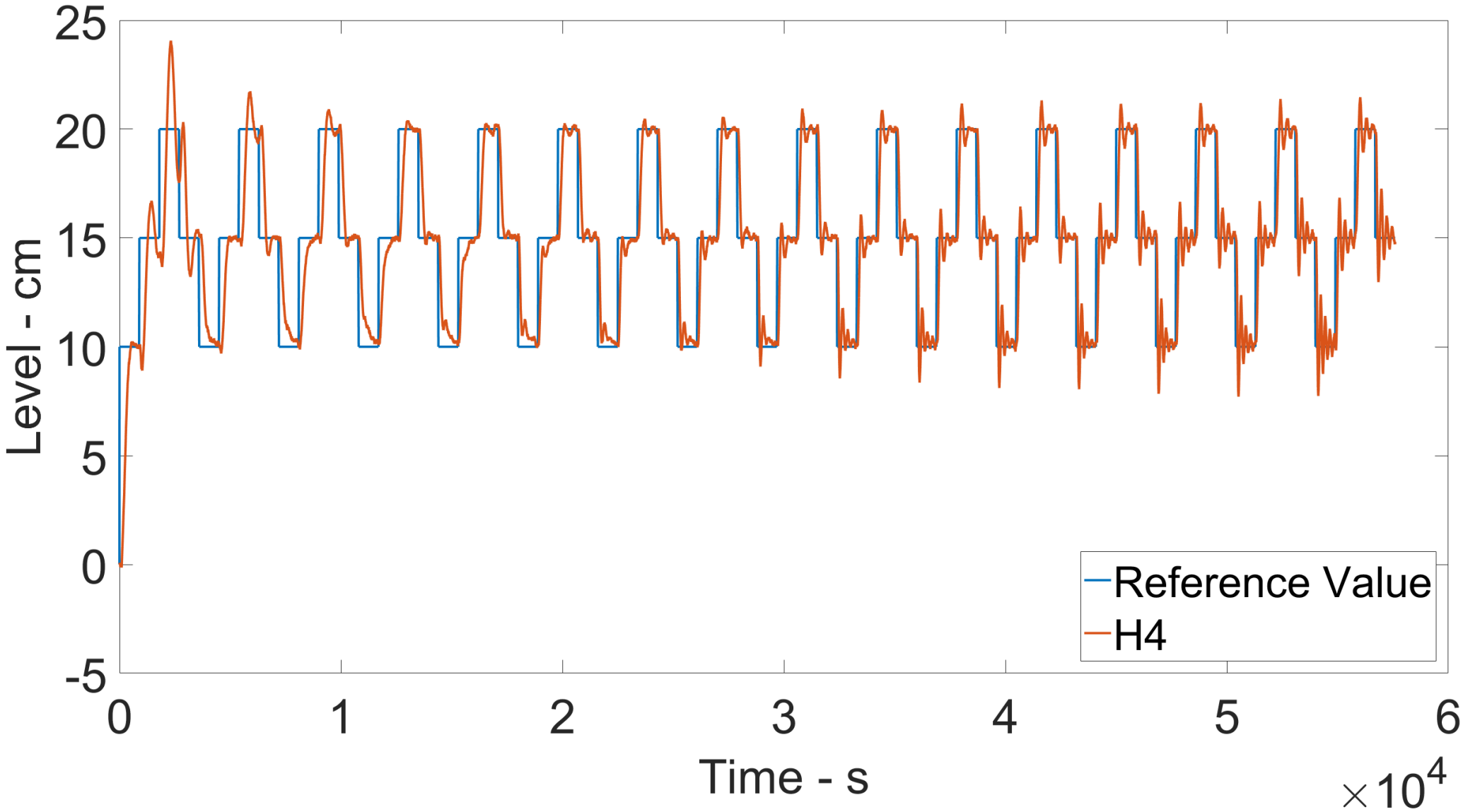
Real RF3 com
Tap = 0.01 e Ts = 0.1s



Real RF3 com
Tap = 0.01 e Ts = 0.5s



Taxa de Amostragem X Taxa de Aprendizado



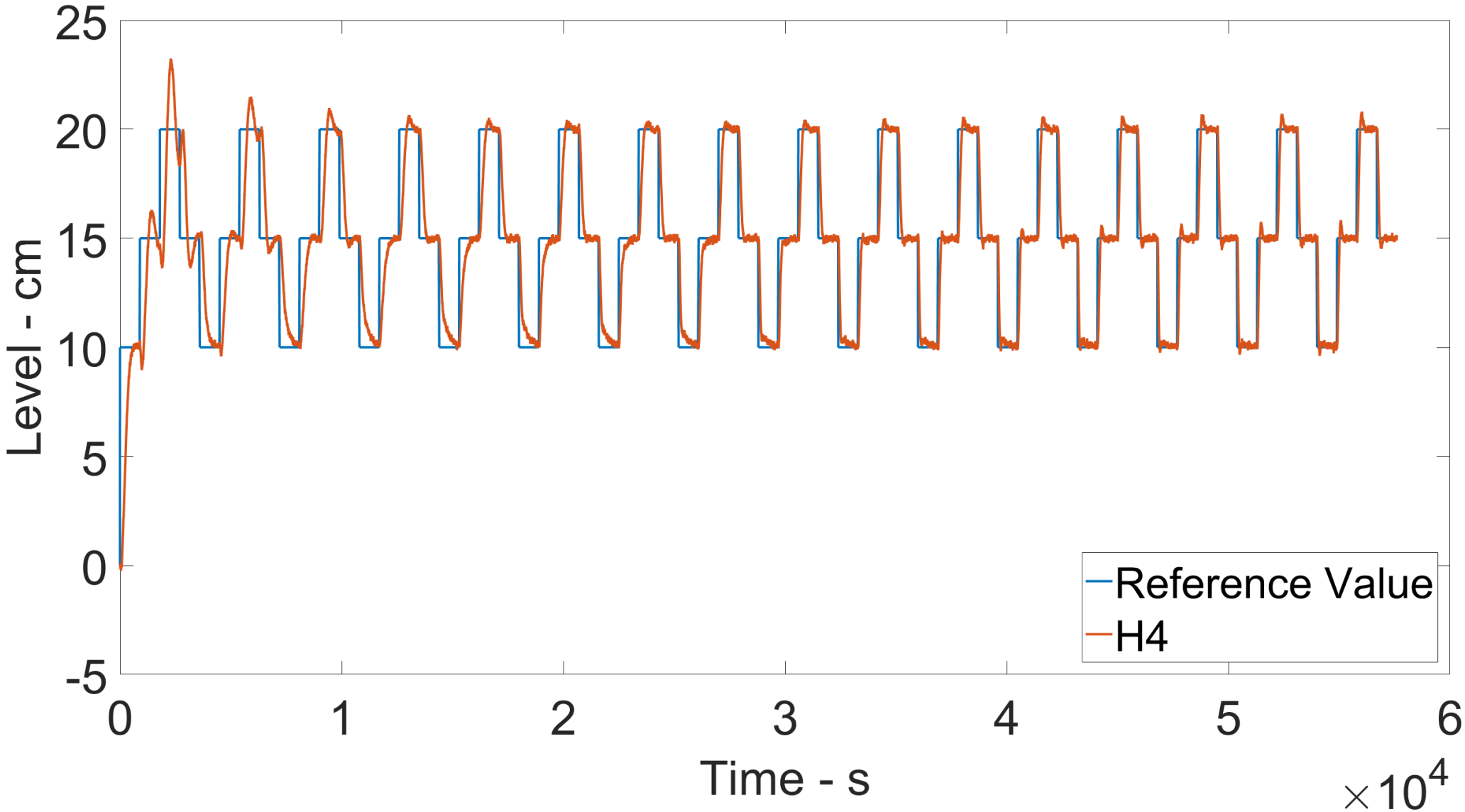
Simulação

RF3

Tap = 0.01

Ts = 1s

Taxa de Amostragem X Taxa de Aprendizado



Simulação

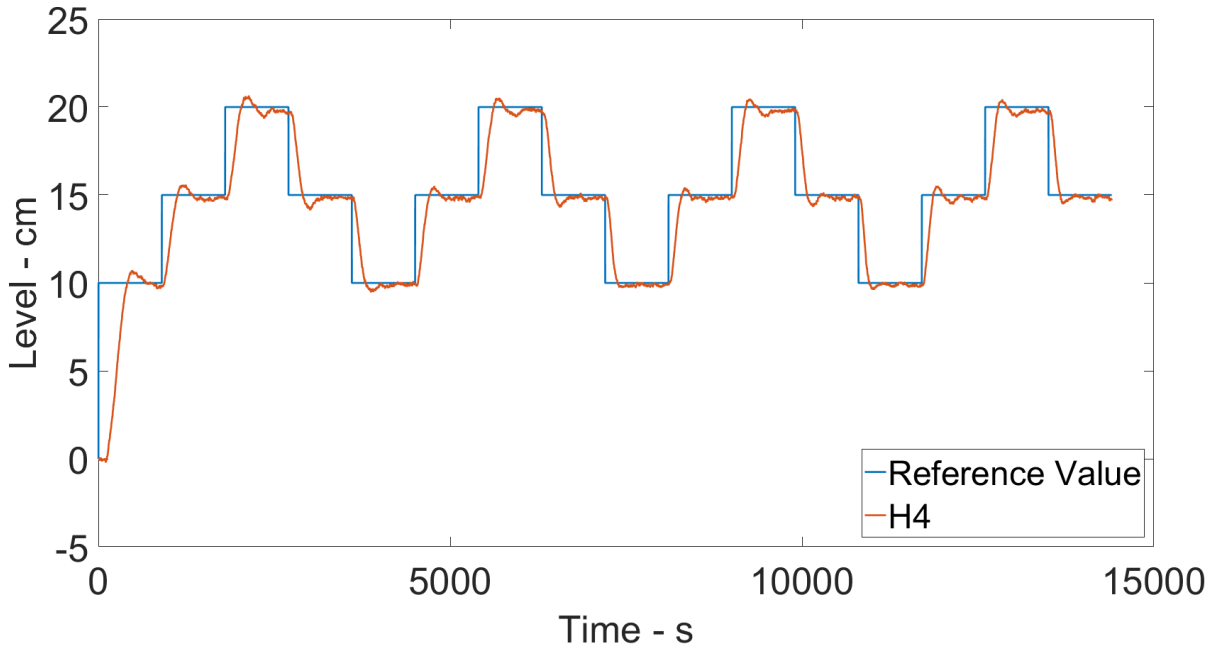
RF3

Tap = 0.001

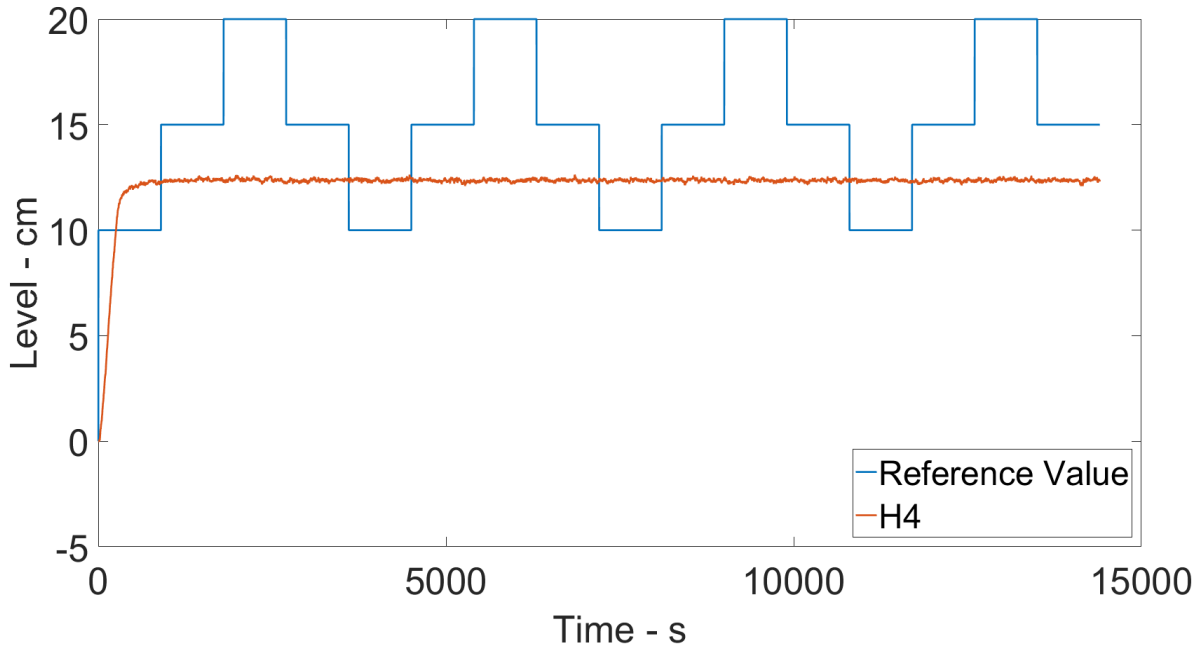
Ts = 0.1s

Sensibilidade a Condições Iniciais

Simulação RF3 com
Tap = 0.01 e Ts = 1s
Condição 1

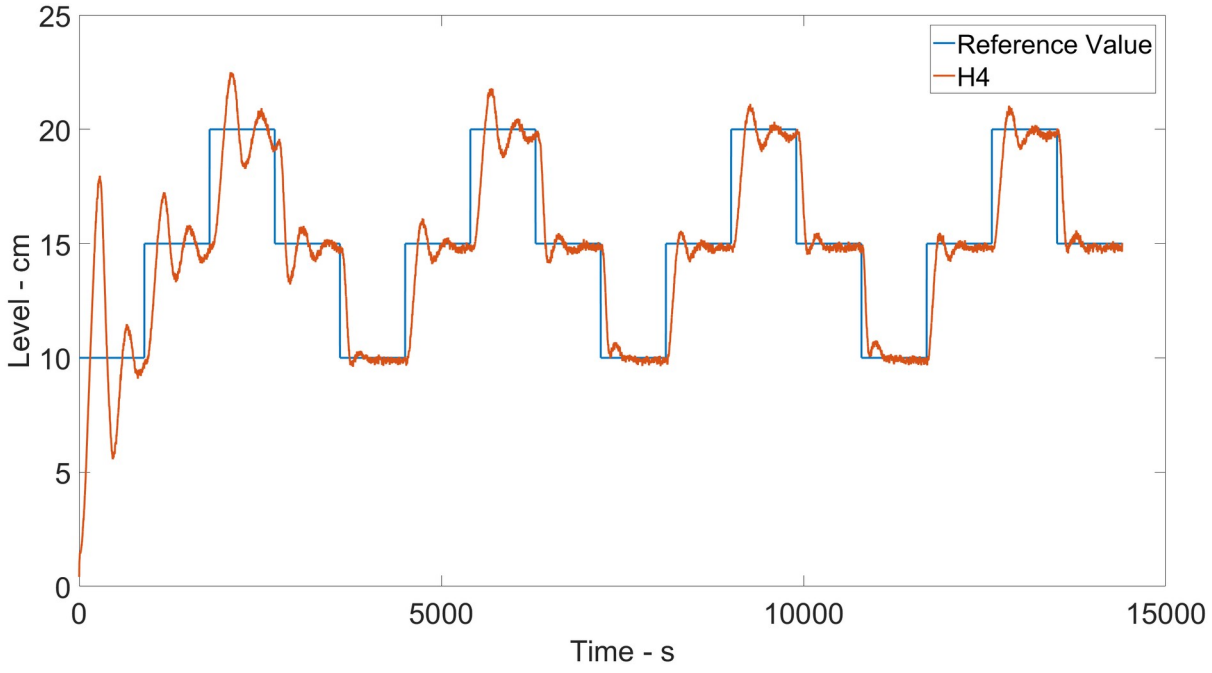


Real RF3 com
Tap = 0.01 e Ts = 1s
Condição 2

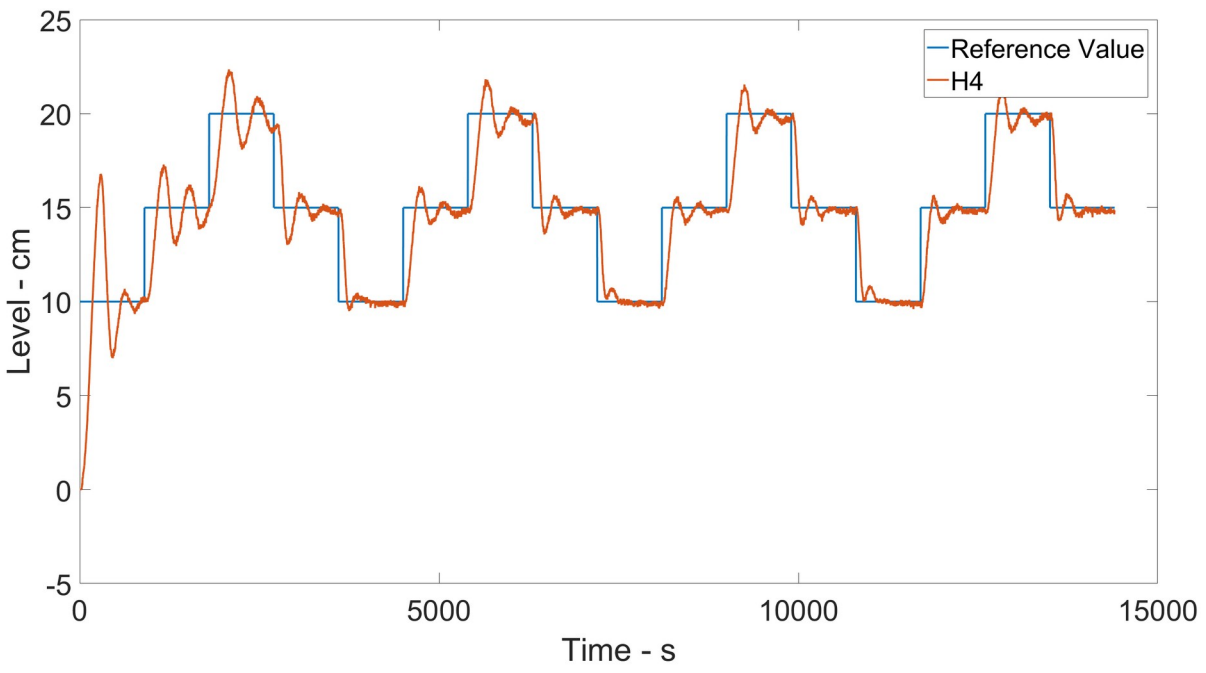


Sensibilidade a Condições Iniciais

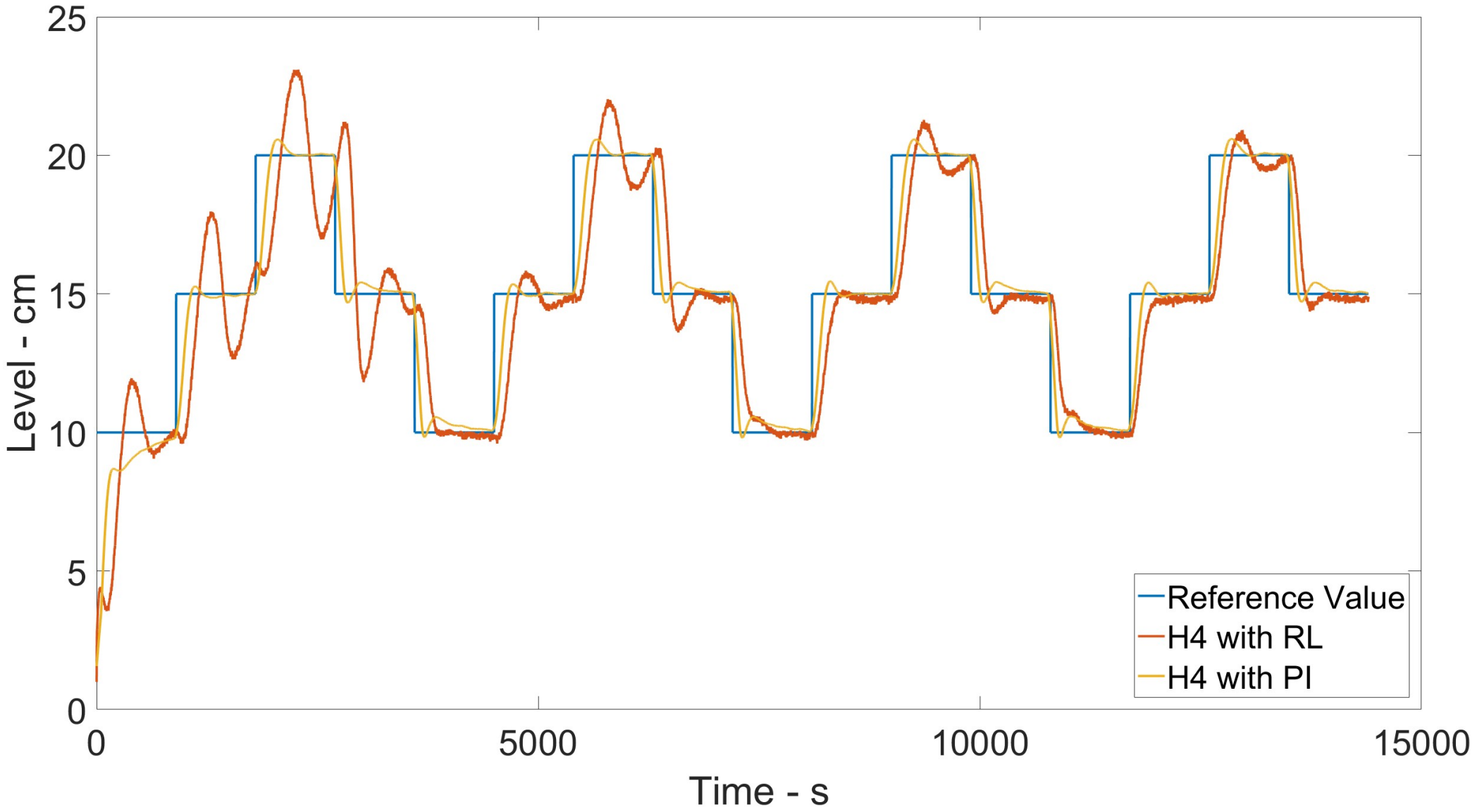
Real RF3 com
Tap = 0.01 e Ts = 0.5s



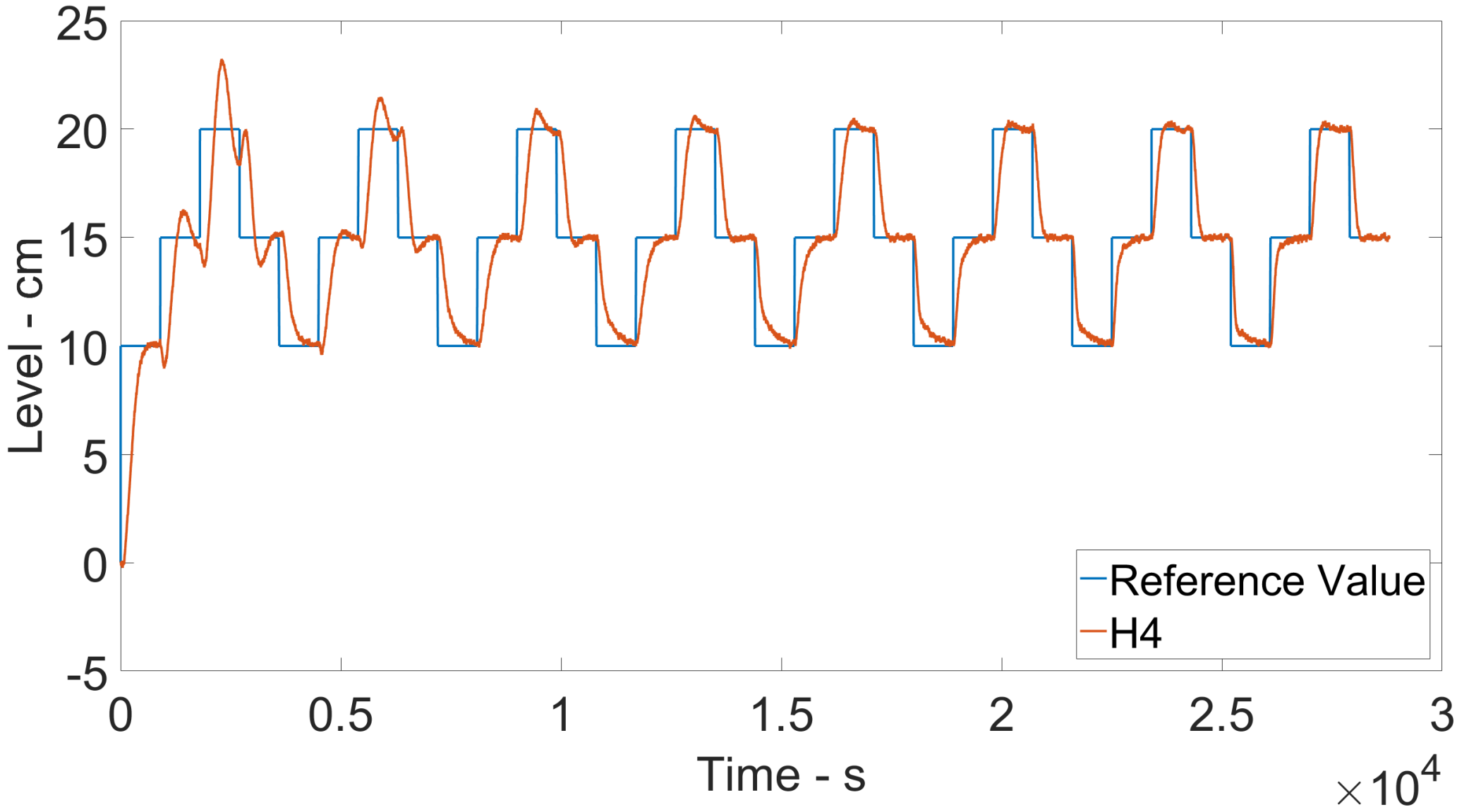
Real RF3 com
Tap = 0.01 e Ts = 0.5s



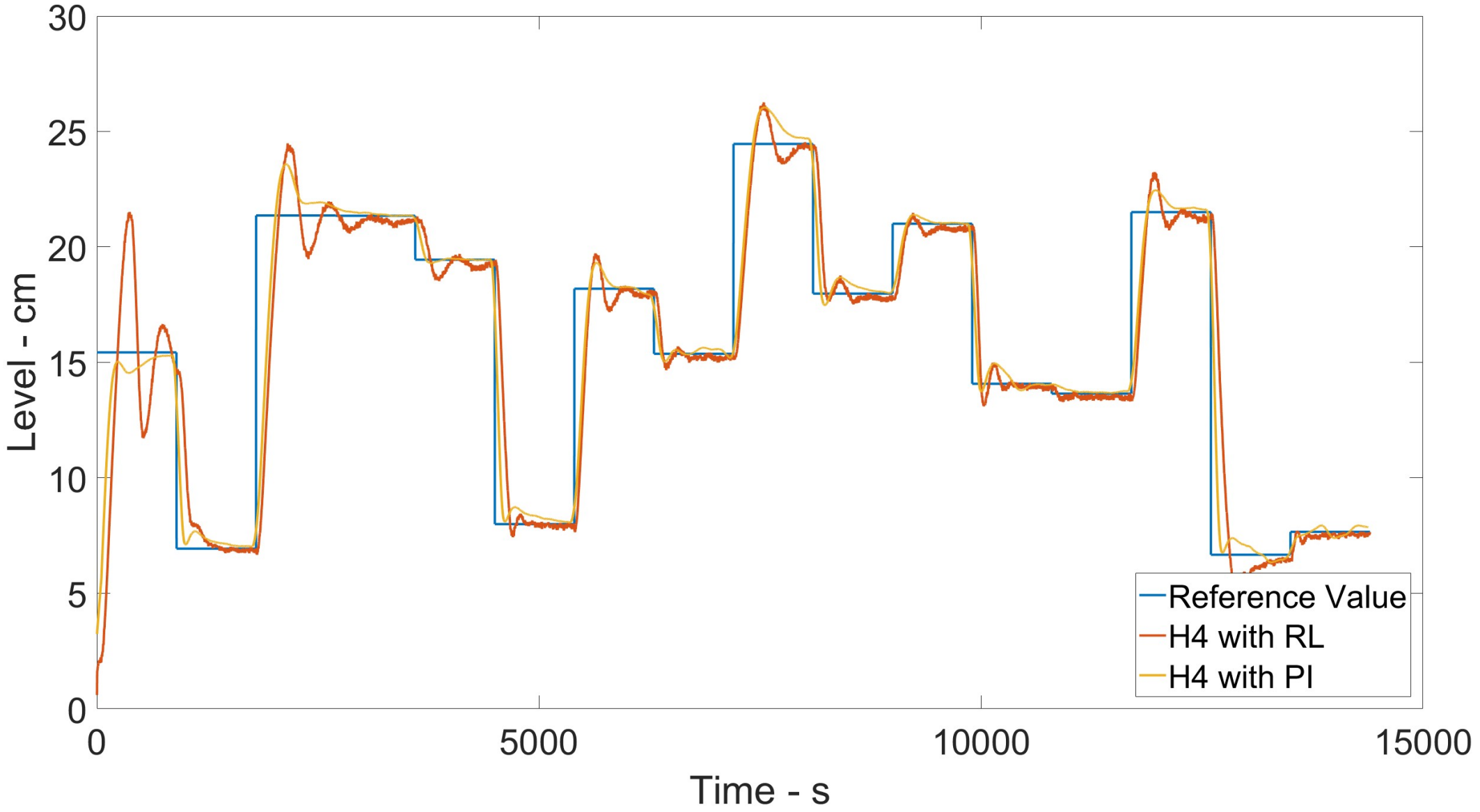
Comparação com um PI 1



Comparação com um PI 1 - Simulação



Comparação com um PI 2



Conclusões

- Controlador funciona dentro de faixas específicas de taxa de amostragem e aprendizado.
- Apresenta certa robustez a condições iniciais e pequenas variações na dinâmica. O algorítmo é estável.
- Demonstra capacidade de aprendizado, característica da adaptibilidade.
- A longo prazo um controlador “mais eficiente” que o de comparação.

Propostas de Trabalhos Futuros

- Estimador de Atraso.
- Variabilidade de Parâmetros.
- Trabalhar em regime de saturação.
- Sistema robusto de aquisição e controle.
- Testar o mesmo controle em outros sistemas.
- Ajuste de taxa de amostragem.

Agradecimentos

- Prof. Adolfo Bauchspiess
- Daniel Vicentin e Vinícius Galvão
- Companheiros do LARA
- Familiares e Amigos

Perguntas?