

# Desenvolvimento de modelo computacional com aplicação do filtro de Kalman para detecção de vazamentos em redes de distribuição de água


- **Data de entrada:**  
16/05/2023
- **Data de aprovação:**  
21/02/2023

## *Development of a computational model with application of the Kalman filter for detecting leaks in water distribution networks*

Rafael Humberto Machado Zerunian<sup>1\*</sup> | Erich Kellner<sup>1</sup>

DOI: <https://doi.org/10.36659/dae.2024.047>

### ORCID ID

Zerunian RHM  <https://orcid.org/0009-0008-4002-0707>

Kellner E  <https://orcid.org/0000-0002-7592-5580>

### Resumo

As perdas reais, causadas sobretudo por vazamentos nas redes de distribuição de água, são um obstáculo importante na gestão de recursos hídricos no meio urbano. Entre as abordagens para o seu combate, está a utilização de métodos probabilísticos que possam simular esse tipo de sistema dinâmico, a exemplo do filtro de Kalman, utilizado no campo científico como uma ferramenta de previsão de estados a partir de dados modelados e dados de sensores cuja medição possui distorções que afetam sua precisão. Este estudo objetivou avaliar a aplicabilidade desse conceito como instrumento operacional e, para isso, foi realizada a modelagem hidráulica do sistema Village Flamboyant, na cidade de São José do Rio Preto (SP). Posteriormente, foram realizadas medições de pressão na rede e, a partir dos dados coletados, foi desenvolvido um algoritmo em Python que permitiu a determinação de valores residuais do filtro de Kalman, demonstrando sua correlação com a ocorrência de vazamentos na rede.

**Palavras-chave:** Perdas reais. Filtro de Kalman. Modelagem hidráulica.

### Abstract

*Real losses, mainly caused by leaks in water distribution networks, are an important obstacle in the management of urban water resources. The approaches to combat it include using probabilistic methods that can simulate this type of dynamic system, such as the Kalman Filter, which the scientific field uses as a state prediction tool based on modeled and sensor data the measurement of which shows distortions that affect its accuracy. This study aimed to evaluate the applicability of this concept as an operational instrument. For that, a hydraulic modeling of the Village Flamboyant system was carried out in the municipality of São José do Rio Preto, São Paulo State. Then, pressure measurements were taken in the network and, from the collected data, an algorithm was developed in Python that enabled the determination of residual values of the Kalman Filter, showing the correlation between them and the occurrence of leaks in the network.*

**Keywords:** Real losses. Kalman Filter. Hydraulic modelling.

<sup>1</sup> Universidade Federal de São Carlos – São Carlos – São Paulo, Brasil

\* **Autor correspondente:** [rafael.zerunian@gmail.com](mailto:rafael.zerunian@gmail.com)

## 1 INTRODUÇÃO

Entre os tópicos relacionados à gestão de sistemas de abastecimento de água, um dos mais importantes é a gestão das perdas (Kingdom; Liemberger; Marin, 2006), uma vez que a água disponível para consumo humano se torna cada vez mais escassa, e os processos de captação, tratamento e distribuição, cada vez mais complexos e desafiadores (Unesco; UN-Water, 2020). Deixar de entregar a água produzida, ou deixar de contabilizá-la, passou a ser um problema com múltiplas facetas, afetando a sustentabilidade ambiental, bem como a saúde financeira dos órgãos de saneamento (Kingdom; Liemberger; Marin, 2006).

As perdas reais, uma das espécies de perdas, referem-se especificamente à água que não chega ao usuário devido a anomalias na rede de distribuição, como vazamentos (Al-Washali; Sharma; Kennedy, 2016). Para o combate a essas perdas, diversas ferramentas, como métodos acústicos e balanço hídrico, têm sido utilizadas com o intuito de detectar vazamentos. Entre elas se incluem modelos matemáticos que estimam a existência e localização dessas fugas de água (Delgado-Aguiñaga; Besançon; Begovich, 2015).

Nesse contexto, o filtro de Kalman é uma das possíveis alternativas para a realização da tarefa, uma vez que sua finalidade é a de estimar o estado de variáveis desconhecidas, a partir de medições observadas ao longo do tempo (Kim; Bang, 2019). Portanto, uma vez que um sistema de distribuição é dotado de variáveis hidráulicas que interagem com a existência de vazamentos, tais como vazão e pressão, a utilização do filtro de Kalman poderia relacionar dados advindos de várias localidades diferentes do sistema e determinar incorreções entre valores medidos em campo e um modelo hidráulico correspondente, funcionando como um estimador de vazamentos (Delgado-Aguiñaga; Besançon; Begovich, 2015).

Assim, este estudo se concentrará nas perdas reais e na detecção de vazamentos, utilizando o filtro de Kalman aplicado a um sistema de abastecimento de água de São José do Rio Preto, em São Paulo.

## 2 OBJETIVO

Este trabalho teve como objetivo principal aplicar o conceito do filtro de Kalman como ferramenta de identificação e localização de vazamentos em redes de distribuição de água. Para isso, este arcabouço teórico foi aplicado em uma situação de abastecimento real em um sistema de abastecimento público de água potável do município de São José do Rio Preto, em São Paulo.

Assim, foram definidos como objetivos específicos: o desenvolvimento do algoritmo computacional, utilizando-se o método do filtro de Kalman, capaz de estabelecer as relações entre os dados coletados e o modelo hidráulico; a aplicação do modelo computacional; e a busca de correlações com dados de campo, a fim de indicar provável localização de vazamentos na rede estudada.

## 3 METODOLOGIA

O filtro de Kalman é um algoritmo que possui como objetivo estimar o estado desconhecido de variáveis, na presença de medições ruidosas e imprecisas, em sistemas dinâmicos lineares. Além disso, esse instrumento matemático permite que seja realizada a predição desse estado em um momento futuro, com base em suas estimativas e correções passadas.

O processo de aplicação do filtro de Kalman impõe a necessidade de assimilação dos dados obtidos, o que significa dizer que, a cada passo de tempo em que o algoritmo é executado, ele realiza uma previsão, que é comparada com a medição. Em seguida, são realizadas uma correção e uma nova previsão para o próximo passo de tempo. Portanto, o filtro de Kalman é composto de três partes principais: um vetor de estado, que geral-

mente contém parâmetros de modelo, bem como variáveis dependentes; um modelo de estimativa; e um modelo de assimilação, que combina informações do modelo de estimativa com dados observados. A diferença entre os modelos é chamada de inovação; e as matrizes de covariância do erro de estado e de covariância do erro de observação são usadas para calcular o peso atribuído à inovação (Ruzza, 2017).

Assim, o filtro de Kalman pode ser expresso através das seguintes equações (Ruzza, 2017):

$$X_{t_i}^f = AX_{t_{i-1}}^u + e_{1t_i} \quad (1)$$

$$P_{t_i}^f = AP_{t_{i-1}}^u A^T + W_{t_i} \quad (2)$$

$$Z_{t_i}^m = MX_{t_i}^f + e_{2t_i} \quad (3)$$

$$K_{t_i} = PF_{t_i} M^T (MPF_{t_i} M^T + R_{t_i})^{-1} \quad (4)$$

$$X_{t_i}^u = X_{t_i}^f + K_{t_i} (z_{t_i}^m - MX_{t_i}^f) \quad (5)$$

$$P_{t_i}^u = (I - K_{t_i} M) P_{t_i}^f \quad (6)$$

Em que:

**X** é o vetor de estados, que representa o estado do sistema, incluindo os parâmetros de modelo, variáveis dependentes e outras observações;

**Z<sup>m</sup>** é o vetor de medições (ou observação);

**M** é o operador de medição que representa a relação entre o vetor de estados e o vetor de medições;

**P** é a matriz covariância de erro de estado;

**R** é a matriz de covariância de erro de medições (observações);

**K** é o ganho de Kalman;

**A** é a matriz de transição linear ou matriz de sensibilidade, que é um operador linear para extrapolar o estado de um passo de tempo para o próximo;

**e<sub>1</sub>** e **e<sub>2</sub>** são ruídos brancos independentes para o modelo de predição e medição (observação), extraídos a partir de distribuições multinormais (gaussianas) com média zero e covariância **W** e **R**, respectivamente;

**t<sub>i</sub>** é o passo de tempo;

**T** (sobrescrito) significa operação de transposição; e

**f** e **u** (sobrescritos) indicam os procedimentos de predição e atualização, respectivamente.

Desse modo, é possível definir que a equação (1) consiste na etapa de predição no tempo  $t_i$ , o modelo de predição de ser executado até a realização de uma nova medição e, então, o vetor de medições  $Z^m$  é transformado a partir do operador de medições, incorporando os erros de medição (observação)  $e_2$  na equação (3). Já na equação (4) é realizado o cálculo do ganho de Kalman, que define o peso das componentes estimadas e medidas, ou seja, quanto do dado obtido pela medição será utilizado no cálculo da nova estimativa.

Por fim, pela equação (5), o vetor de estados é atualizado com a soma ponderada da estimativa e da inovação, sendo também atualizada a covariância erro de estado pela equação (6), que deverá ser inferior à covariância estimada anteriormente pela equação (2).

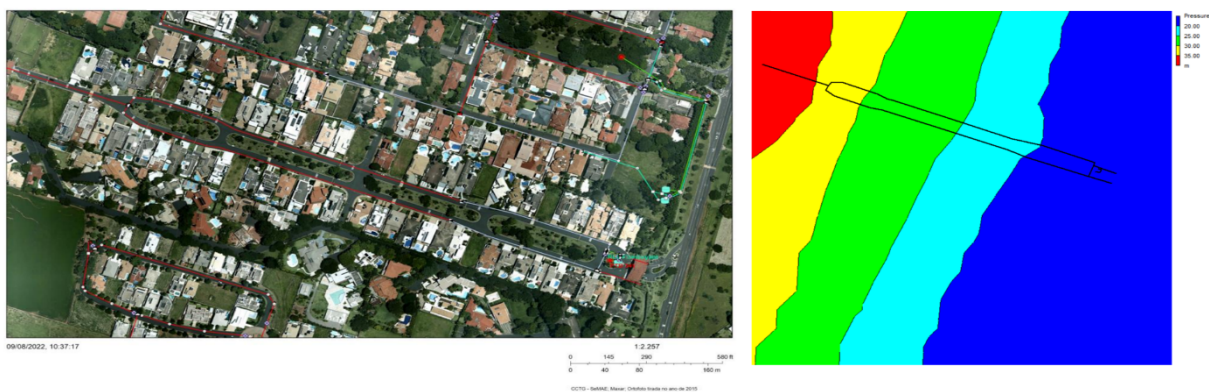
A metodologia deste estudo foi dividida em duas partes para verificar a eficácia do filtro de Kalman na detecção e localização de vazamentos. A primeira etapa envolveu a seleção de um sistema de distribuição real em São José do Rio Preto (SP) para testes, além do levantamento de dados cadastrais e operacionais, bem como a simulação hidráulica do sistema escolhido. A segunda fase focou na criação de um algoritmo para aplicar a metodologia aos casos simulados; na coleta de dados de pressão com *dataloggers*; e na realização de simulações de vazamentos em pontos específicos da rede de abastecimento. Os

dados coletados foram utilizados no modelo para verificar o desempenho da metodologia nesse cenário de abastecimento.

### 3.1 Criação do modelo no EPANET

Tendo em vista que o objeto de análise primordial desta pesquisa é a aplicação do conceito do filtro de Kalman em um sistema de abastecimento de água real, as simulações para a prototipagem do software foram baseadas em informações de um sistema de abastecimento em São José do Rio Preto gerenciado pelo Serviço Municipal Autônomo de Água e Esgoto (SeMAE).

A seleção do sistema levou em consideração critérios como o comprimento da rede, ausência de interligações entre sistemas, facilidade de acesso ao hidrômetro e a fidelidade do cadastro. O objetivo foi minimizar a influência de variáveis desconhecidas e maximizar a representatividade dos dados coletados. Entre os sistemas avaliados, Harmonia e Village Flamboyant se destacaram. O estudo focou no sistema Village Flamboyant (Figura 1) devido à sua topografia com maior declividade, permitindo maior variação de elevação e zonas de pressão mais bem definidas, conforme também se verifica na figura.



**Figura 1** – Ortofoto do sistema Flamboyant (à esquerda) e zonas de pressão (à direita).

Fonte: SeMAE, 2015; elaboração própria, 2023.

Para simular o sistema Village Flamboyant no EPANET, foi identificado que ele tem 68 ligações, 1518 m de rede, um poço “PTB-341” com bomba Leão R20A-12 (instalada a cerca de 85 m) e vazão de 22,8 m<sup>3</sup>/h, um reservatório elevado de 16 m de altura máxima e 10 m de altura mínima, com 4 m de diâmetro, demanda média de 7,2 m<sup>3</sup>/h e perfil de demanda residencial padrão. Depois de inserir os dados no EPANET, um modelo foi gerado e exportado para um arquivo “.INP” como demonstrado na Figura 2. As demandas foram então distribuídas em cada nó, através

da divisão do valor médio micromedido total de forma ponderada em relação ao comprimento da rede conectada a cada nó, como se verifica na Figura 3.

O arquivo, já com as demandas distribuídas, foi copiado para um segundo arquivo no qual seriam simuladas medições dos *dataloggers*, criando-se, posteriormente, uma variação gaussiana aleatória em torno de um erro de 0,01 no código em *Python*, representando as perturbações de sinal/ruídos do transdutor de pressão utilizado.

Para retratar o perfil de consumo da população do bairro, foram utilizados os dados horários de vazão da distribuição de agosto de 2020 a julho de

2021 do reservatório e, através da média, traçado o seguinte padrão de demanda (*pattern*) para o EPANET (Figura 4).

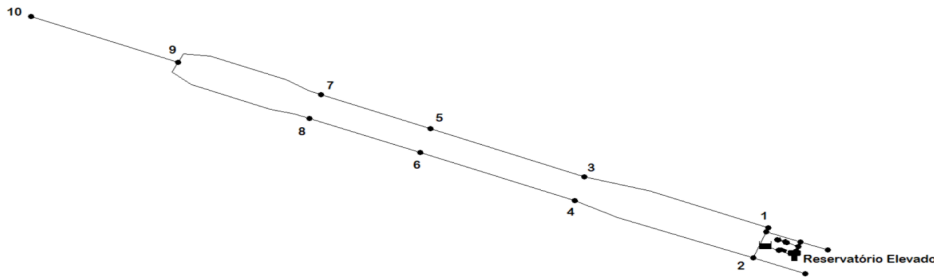


Figura 2 – Modelo EPANET do sistema Flamboyant.

										Micromedidação média de 12 meses (m <sup>3</sup> /mês)		5185.65
										Micromedidação média de 12 meses (m <sup>3</sup> /h)		7.20
										Demanda por metro de rede (m <sup>3</sup> /h)		0.0047603
[PIPES]	ID	Node1	Node2	Length	Diameter	Roughness	MinorLoss	Status		Nós	Comprimento Atribuído(m)	Demanda Atribuída (m <sup>3</sup> /h)
	3	13	12	3.07	100	150	0	Open ;		1	96.56	0.459651873
	4	12	11	18.92	100	150	0	Open ;		2	132.355	0.630045812
	5	12	17	33.19	50	150	0	Open ;		3	170.455	0.811412178
	6	11	1	8.95	75	150	0	Open ;		4	169.375	0.806271085
	7	11	2	28.18	75	150	0	Open ;		5	133.885	0.637329028
	8	2	18	54.8	50	150	0	Open ;		6	134.615	0.640804027
	9	1	3	184.17	75	150	0	Open ;		7	130.82	0.622738794
	10	2	4	181.73	75	150	0	Open ;		8	133.83	0.637067213
	11	7	9	150.61	50	150	0	Open ;		9	227.935	1.085032618
	12	8	9	155.45	50	150	0	Open ;		10	74.905	0.356568181
	13	9	10	149.81	50	150	0	Open ;		11	28.025	0.133406625
	14	3	5	156.74	50	150	0	Open ;		12	27.59	0.131335907
	15	5	7	111.03	50	150	0	Open ;		13	1.535	0.007307018
	16	4	6	157.02	50	150	0	Open ;		14	0	0
	17	6	8	112.21	50	150	0	Open ;		15	0	0
	19	14	13	0	100	150	0	Open ;		16	3.56	0.016946569
	20	16	19	7.12	100	150	0	Open ;		17	16.595	0.078996715
										18	27.4	0.130431455
										19	3.56	0.016946569
	Comprimento Total (m)			1513								

Nós relacionados ao trecho de rede do terminal de limpeza

Figura 3 – Memória de cálculo das demandas por metro de rede atribuído aos nós.

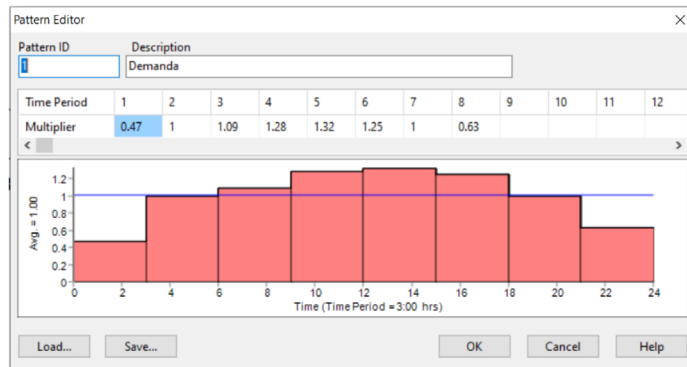


Figura 4 – Ortofoto do sistema Village Flamboyant.

Com o objetivo de simular um vazamento em um dos nós do bairro, foi criado um nó adicional (nó 21) após o nó 10, e nele foi configurado um emissor (*emitter*) apenas no arquivo contendo as medições simuladas, com um coeficiente de valor 0.5.

Um *emitter* no EPANET permite a simulação, entre outros cenários, de uma perda real em um nó da simulação. Segundo o EPANET 2.0 – Manual do Usuário (Rossman, 2000), “a vazão que sai do dispositivo emissor varia como uma função da pressão disponível no nó”, e tal fenômeno se expressa pelo produto do coeficiente de vazão pela pressão no nó, elevada a um expoente. É justamente esse expoente que será inserido no nó 21 para representar a intensidade desse vazamento e, apesar de o valor ter uma variação desconhecida, o coeficiente de um vazamento comumente varia entre 0,5 (pequeno vazamento) e 2 (rompimento de rede).

### 3.2 Desenvolvimento do algoritmo e do código em Python

Para aplicar o filtro de Kalman à questão das perdas reais em redes de abastecimento de água, foi necessário converter os conceitos matemáticos para uma linguagem de programação. A linguagem Python foi escolhida devido à sua comunidade ativa, ampla biblioteca para operações matemáticas, curva de aprendizado rápida e uso frequente em programas de geoprocessamento, como Qgis e ArcGis, que têm módulos para modelagem hidráulica com o motor do EPANET.

#### 3.2.1 Compatibilizando dados EPANET e Python

Após extensa pesquisa, foi possível encontrar um trabalho bastante detalhado que, através da interface de programação de aplicações (API) *EpanetTools* (Pathirana, 2015), possibilita a realização da tarefa de portar os dados entre as duas plataformas. Apesar de essa etapa não possuir ligação direta com o objeto deste trabalho, a inexistência de tal ferramenta poderia desenco-

rajar quem desejasse criar soluções baseadas em *Python*. A API citada tira proveito do módulo *Epanet Programmers Toolkit* (USEPA), que expõe funções do *CORE* do programa para serem utilizadas externamente. A API, então, acessa essas funções e importa os dados para o *Python*. Por conta da sua facilidade de utilização, para essa etapa foi utilizada a API *EpanetTools* com o seguinte código, que permite a importação dos arquivos “.INP” gerados pelo EPANET:

```
from epanettools.epanettools import
EPANetSimulation, Node
es=EPANetSimulation(Caminho_do_arquivo)
es.run()
```

Durante a etapa de prototipagem, o modelo do EPANET gerado para o sistema de abastecimento foi utilizado como Modelo do Processo. Além disso, o mesmo modelo teve seus valores modificados por um erro teórico aleatório com distribuição normal sobre a acurácia de 0,1, permitindo a verificação do desempenho dos valores estimados pelo filtro em relação aos valores simulados pelo modelo do EPANET.

#### 3.2.2 Estrutura do algoritmo para prototipagem

(1) Carregue os dados da simulação do EPANET referente aos 24 valores de pressão de 10 nós do sistema de abastecimento de água (Modelo do Processo).

(2) Reutilize os dados carregados aplicando uma distribuição normal do erro do modelo para representação dos valores medidos ( $Z^m$ ).

(3) Inicialização:

(3.1) Defina o Estado Inicial baseado nos dados iniciais do EPANET ( $X_{t_i}^u$ )

(3.2) Defina a Covariância do Erro Inicial ( $P_{t_i}^u$ )

(3.3) Defina a Matriz de Transição de Estados (A), que será uma matriz identidade ( $A_g$ ), uma vez que



a simulação do EPANET já define a dinâmica de transição em cada passo de tempo diretamente pelo resultado da própria simulação hidráulica.

(3.4) Defina a Matriz de Medição (M), que será uma matriz identidade ( $M_b$ ), uma vez que o parâmetro estimado é o mesmo parâmetro medido, com relação, portanto, de 1:1. Tecnicamente, poderia ser ignorada, mas sua inclusão facilita a legibilidade do processo. A Matriz teria relação diferente em caso de medições indiretas, como a medição de pressão para verificação do nível de um reservatório cilíndrico, por exemplo.

(3.5) Defina a Matriz de Covariância do Erro da Estimativa (P), inicializada com um valor (P8) e atualizada em cada etapa.

(3.6) Defina a Matriz de Covariância do Erro do Modelo (W), que será uma matriz identidade multiplicada pelo Erro.

(3.7) Defina a Matriz de Covariância do Erro da Medição (R), que será uma matriz identidade multiplicada pelo Erro.

(4) Predição:

(4.1) Estado a priori:  $X_{t_i}^f = AX_{t_{i-1}}^u + e_{1t_i}$

(4.2) Covariância do erro a priori:  $P_{t_i}^f = AP_{t_{i-1}}^u A^T + W_{t_i}$

(5) Atualização:

(5.1) Calcule o Residual:  $Z_{t_i}^m = MX_{t_i}^f + e_{2t_i}$

(5.2) Calcule o Ganho de Kalman:

$$K_{t_i} = PF_{t_i} M^T (MPF_{t_i} M^T + R_{t_i})^{-1}$$

(5.3) Atualize a estimativa do estado a posteriori:

$$X_{t_i}^u = X_{t_i}^f + K_{t_i} (z_{t_i}^m - MX_{t_i}^f)$$

(5.4) Atualize a covariância do erro do estado a posteriori:  $P_{t_i}^u = (I - K_{t_i} M) P_{t_i}^f$

(6) Residual Normalizado:

(6.1) Normalize os residuais usando os residuais brutos de cada etapa:  $Y^n = Y^b / Z$

### 3.2.3 Tratamento dos dados em Python

Para que os valores obtidos pudessem ser utilizados de forma inteligível durante as etapas do loop do filtro, os dados foram inseridos em variáveis adequadas.

```
import numpy as np
import matplotlib.pyplot

from epanettools.epanettools import
EPANetSimulation, Node

h = 24 # Número de horas
n = 10 # Número de nós de monitoramento
#Valores de pressão na Simulação do EPANET de 8
nós em 24 horas
S = [[] for _ in range(n)]
M = [[] for _ in range(n)]
#Carregando Simulação
es=EPANetSimulation('Caminho_do_arquivo.inp')
es.run()
# Passo de tempo do modelo
tstep = list(es.network.time)
# Parâmetro utilizado do modelo do EPANET
p = Node.value_type['EN_PRESSURE']
for node in range(n):
nid = node + 1
results = es.network.nodes[nid].results[p]
hour = 0
for simHour in range(len(tstep)):
t = (tstep[simHour]) / 3600
if t.is_integer() and hour < 24:
S[node].append(round(results[int(t)], 2))
M[node].append(round(np.random.
normal(results[int(t)], 0.1), 2))
hour += 1
S = np.array(S)
M = np.array(M)
A = np.eye(n) # Matriz de Transição de Estados
Z = np.eye(n) # Matriz de Observação
```

```

W = np.eye(n) * 0.001 # Matriz de covariância do
ruído de processo
R = np.eye(n) * 0.01 # Matriz de covariância do ruído
de Medição
B = np.eye(n) #Matriz de controle
X = S[:, 0] # Estado inicial
P = np.eye(n) * 1.0 # Covariância inicial
estimates = np.zeros((n, h))
residuals = np.zeros((n, h))

```

### 3.2.4 Filtro de Kalman (loop predict/update)

Neste bloco foi criado o laço de repetição para a realização dos cálculos iterativos do filtro de Kalman explicados anteriormente, de forma que, a cada passo de tempo, os valores do ganho de Kalman (k) sejam atualizados e aplicados para a geração da previsão seguinte, além do cálculo do valor residual resultante.

```

for t in range(h):
#ENTRADA DE CONTROLE.
u = S[:, t] - S[:, t - 1] if t > 0 else np.zeros(n)
#PREDIÇÃO.
x_pred = A @ X + B @ u
p_pred = A @ P @ A.T + W
#ATUALIZAÇÃO:
residual = M[:, t] - Z @ x_pred
residual_normalized = np.round(np.abs(residual) /
M[:, t], 4)
residual_covariance = Z @ p_pred @ Z.T + R
K = p_pred @ Z.T @ np.linalg.
inv(residual_covariance)
# Atualizando o estado (X) e a matriz de covariância
(P) do sistema
X = x_pred + K @ residual
P = (np.eye(n) - K @ Z) @ p_pred
# Armazenando as estimativas do estado para cada
instante de tempo
estimates[:, t] = X

```

```

# Armazenando os resíduos normalizados para cada
instante de tempo
residuals[:, t] = residual_normalized
# Transformando os resíduos em uma matriz
residuals_matrix = np.array(residuals)

```

### 3.2.5 Plotagem dos dados

O próximo bloco de código tem como função apenas “plotar” os dados de forma gráfica, permitindo uma melhor visualização do efeito produzido pelo filtro. Nesta etapa do algoritmo, os dados ainda estão sendo demonstrados através de uma consulta em um nó específico (nó 10).

```

fig, chart = matplotlib.pyplot.subplots()
fig, chart = matplotlib.pyplot.subplots()
chart2 = chart.twinx()
med = chart.plot(M[9], color='black',
label='medição')
est = chart.plot(estimates[9],
color='blue',label='estimativas')
res = chart2.plot(residuals[9], color='red',
label='residual')
leg = sim+est+res
labs = [l.get_label() for l in leg]
chart.legend(leg, labs, bbox_to_anchor=(0.5, 1),
loc='upper center')
chart.set_xlabel('Horas')
chart.set_ylabel('Pressão(mca)')
chart2.set_ylabel('Residual')
matplotlib.pyplot.show()

```

### 3.3 Análise dos dados

Para verificação da capacidade de ajuste do filtro, foi gerado um gráfico onde é possível verificar a relação entre o filtro e os valores simulados e os valores da simulação com o ruído correspondente à medição de sensores de pressão. Contudo, para que seja possível comparar a dimensão da variação entre o previsto e o medido neste e nos demais nós, é preciso normalizar este valor, operação que



é realizada através da divisão da diferença do estimado menos o medido pelo estimado. Este dado é chamado de residual e o monitoramento desta variável poderá indicar uma aproximação do local com maior probabilidade de vazamentos.

Conforme fica claro na Figura 5, o filtro tem uma boa capacidade de ajuste, fornecendo bons resul-

tados mesmo com a adição do ruído gaussiano nas medições, possuindo valores residuais próximos de 0,16 no máximo. Na sequência, o programa foi executado novamente, porém agora com o *emitter* ativado entre 14 e 15 horas no arquivo que simula as medições em campo. A Figura 6 demonstra o efeito deste vazamento na execução do filtro de Kalman.

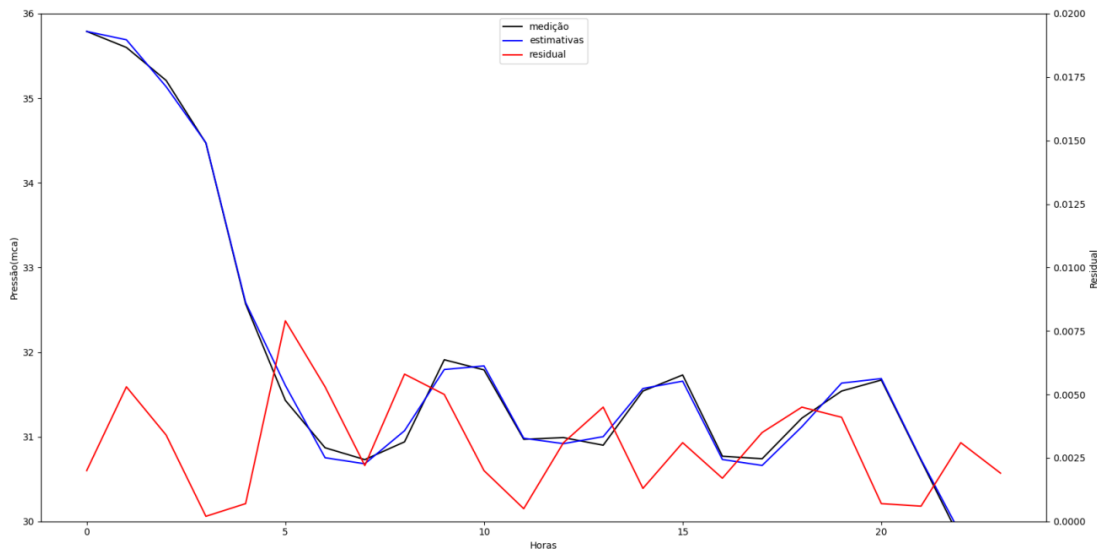


Figura 5 – Filtro de Kalman sem vazamentos.

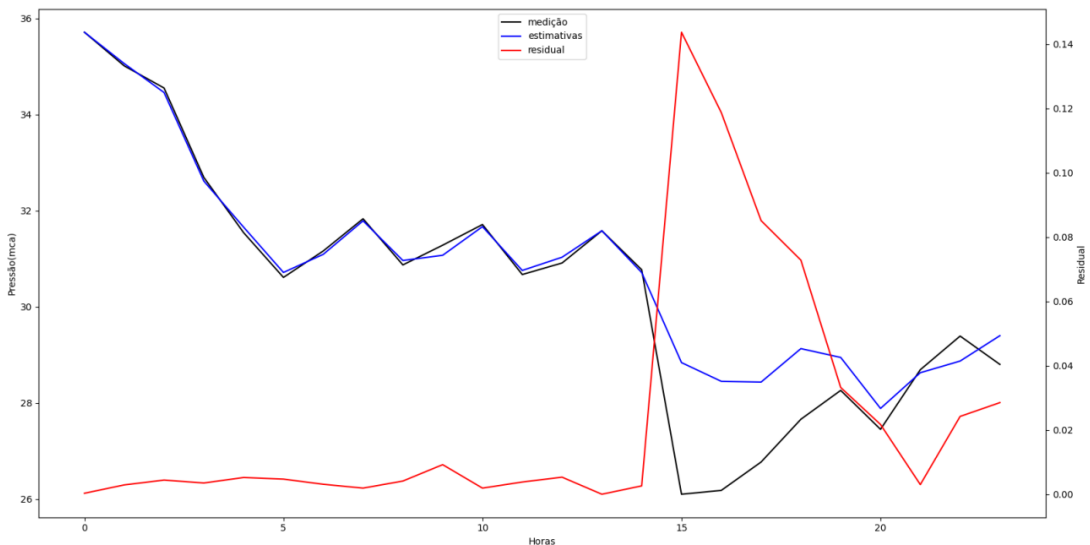


Figura 6 – Filtro de Kalman com vazamento.

Como se verifica, a queda de pressão causada pelo vazamento impacta de forma inversa no residual, que ultrapassa o valor de 0,14, exigindo uma correção mais agressiva do filtro de Kalman, o qual tende a se aproximar mais do modelo cujo erro é mais baixo (0,001) do que da medição

(0,01). Portanto, a partir da aplicação teórica do algoritmo é que se torna verossímil a hipótese de aplicação simultânea em todos os nós, que deverão apresentar maiores distanciamentos entre a simulação e a medição quanto mais próximos do evento causador do distúrbio.

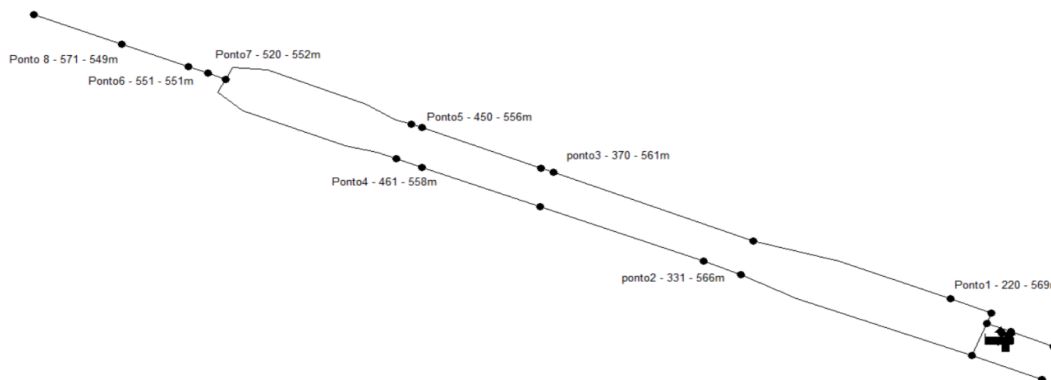


Figura 7 – Novo cenário EPANET.

### 3.4 Aplicação do método em dados reais

Utilizando o produto final da etapa de desenvolvimento e testagem do software, foi planejada e realizada a coleta de dados no sistema Village Flamboyant. Para isso, foram definidos os pontos de medição exatos, e o modelo do EPANET foi ajustado para expressar corretamente a exata posição dos nós, repetindo-se o processo de alocação de

demandas nesse cenário, como demonstrado na Figura 7.

#### 3.4.1 Montagem e programação dos dataloggers

Para a realização da tarefa de coleta e armazenamento dos dados em campo, foram escolhidos os componentes demonstrados na Figura 8, utilizados em cada um dos oito equipamentos montados:



Figura 8 – ESP32 (TTGO Lilygo T3 - v1.6.6, com leitor de cartão microSD) (à esquerda); invólucro IP68 (no meio); e transdutor de pressão 0,5 a 4,5 v com range de 0 a 1,2 Mpa (à direita).

Além dos componentes acima, foram utilizados cabos para conexão dos pinos, cartões de memória (microSD) de 2 Gb de capacidade e baterias de íons de lítio modelo 18650 3,7 v 2600 mAh.

Cada unidade dos oito equipamentos teve seus terminais e cabos soldados e revestidos com tubo isolante termorretrátil, a fim de garantir a segurança do equipamento contra curtos e interferências nas leituras. Os sensores de pressão tiveram seus cabos soldados nos pinos 3.3 v, GND e GPIO34 para a leitura de sinal, as baterias foram conectadas através da porta de alimentação JST nativa desse modelo do ESP32, e foi instalado um botão na GPIO23 com a finalidade de ligar e desligar o sistema WiFi nativo da placa, a fim de preservar a capacidade da bateria.

Cada *datalogger* recebeu um pequeno arquivo HTML (*index.html*), que serviu como interface para iniciar/parar a coleta, apagar dados existentes no cartão de memória e verificar *in loco* uma tabela com as leituras já realizadas. Esta página foi servida por meio de um *webserver* simples na programação do ESP32, que foram programadas com a utilização da plataforma ARDUINO-IDE. Concluída a montagem dos *dataloggers*, os equipamentos foram instalados em oito pontos da rede em cavaletes e caixas padrão que permitiam a instalação com maior facilidade de acesso. Para isso, foi realizada a substituição do cotovelo antes do hidrômetro por um “tê” onde foi instalado o transdutor de pressão. Após a fixação de cada equipamento, este era acessado pela rede WiFi e era iniciado o processo de coleta de dados a cada 1 minuto.

### 3.4.2 Coleta de dados e aplicação no algoritmo Python

Os equipamentos foram instalados entre 15 e 16 horas do dia 13 de março de 2023 e foram removidos no mesmo horário no dia 15 de março de 2023. As leituras consideradas para o estudo foram as compreendidas entre 00 e 23:59 horas do dia 14 de março de 2023, ou seja, um ciclo de 24 horas. No dia 14 de março de 2023, entre 15 e

16 horas foi realizada a abertura do terminal de limpeza da rede, localizado na cota mais baixa do bairro, a fim de simular um vazamento e gerar uma perturbação correspondente nas leituras dos dispositivos instalados.

Posteriormente, após a remoção dos equipamentos, os cartões de memória foram descarregados; e os arquivos com as leituras foram utilizados para a consolidação da pressão média em cada ponto, em cada horário, tendo sido tais dados inseridos no algoritmo no lugar dos valores de medição simulados.

No código, é possível verificar as listas com os valores de pressão média horária para cada ponto obtidos pelos *dataloggers*. Além disso, foi utilizada a biblioteca “Panda” para a criação de uma tabela “.xlsx” com os valores residuais de cada ponto em cada horário, a qual depois foi utilizada como fonte de dados para a criação do mapa temático. Subsequentemente, os dados coletados em campo foram alocados na lista definida anteriormente como “M”. Os dados foram inseridos no loop iterativo do filtro de Kalman, que retornou os valores residuais inseridos no arquivo “residual.xlsx” de forma tabelada.

```
import numpy as np
from epanettools.epanettools import
EPANetSimulation, Node
h = 24 # Número de horas
n = 8 # Número de nós de monitoramento
S = [[] for _ in range(n)]
es=EPANetSimulation('caminho_do_arquivo.inp')
es.run()
tstep = list(es.network.time)
p = Node.value_type['EN_PRESSURE']
for node in range(n):
    nid = node + 1
    results = es.network.nodes[nid].results[p]
    hour = 0
    for simHour in range(len(tstep)):
```

```

t = (tstep[simHour]) / 3600
if t.is_integer() and hour < 24:
S[node].append(round(results[int(t)], 2))
hour += 1
S = np.array(S)
M = np.array([[18.44, 18.17, 18.07, 17.92, 17.79,
17.4, 16.92, 15.91, 15.4, 15.61, 15.49, 15.09,
16.2, 14.79, 15.71, 15.2, 15.8, 17.95, 17.34,
18.44, 18.29, 17.89, 16.65, 17.82], [21.92, 21.76,
21.57, 21.43, 21.26, 20.88, 19.63, 17.95, 18.22,
19.75, 17.62, 18.42, 18.24, 20.51, 16.26, 17.2,
17.41, 21.37, 19.07, 21.78, 21.55, 21.28, 20.98,
21.65], [26.48, 26.29, 26.18, 26.02, 25.44, 22.03,
21.76, 25.43, 21.74, 23.49, 21.86, 22.1, 21.55,
24.39, 22.91, 20.05, 21.09, 25.77, 25.3, 26.46,
26.22, 25.84, 25.56, 26.28], [29.28, 29.2, 29.12,
28.99, 28.76, 28.24, 28.71, 27, 24.75, 26.55,
27.16, 26.21, 28, 28.39, 26.69, 23.41, 24.34,
28.63, 26.66, 29.22, 29.08, 28.51, 28.41, 29.27],
[30.9, 30.69, 30.52, 30.44, 30.35, 30.01, 29.66,
28.96, 30.27, 27.55, 29.63, 29.65, 27.7, 31.69,
31.01, 25.09, 26.84, 30.18, 29.58, 30.4, 30.65,
30.3, 30.01, 30.23], [38.13, 38.03, 37.89, 37.78,
37.62, 37.03, 37.05, 36.05, 34.33, 35.9, 32.48,
30.93, 32.98, 37.94, 32.37, 28.27, 35.33, 37.47,
36.79, 37.59, 37.56, 37.44, 37.18, 36.66], [39.32,
39.44, 39.29, 39.06, 38.9, 38.32, 38.3, 33.85,
32.42, 33.75, 32.17, 31.31, 32.77, 31.81, 31.34,
27.52, 34.34, 39.06, 38.38, 39.36, 38.85, 39.23,
38.52, 39.15], [39.54, 39.41, 39.32, 39.16, 39.06,
38.48, 38.56, 34.21, 34.58, 37.01, 34.86, 39.56,
38.7, 36.82, 35.35, 29.29, 33.06, 38.81, 38.32,
39.06, 39.29, 38.88, 38.72, 39.14]])
F = np.eye(n) # Matriz de Transição de Estados
H = np.eye(n) # Matriz de Observação
Q = np.eye(n) * 0.0001 # Matriz de covariância do
ruído de processo
R = np.eye(n) * 0.01 # Matriz de covariância do ruído
de Medição
B = np.eye(n) # Matriz de controle
X = S[:, 0] # Estado inicial
P = np.eye(n) * 1.0 # Covariância inicial

```

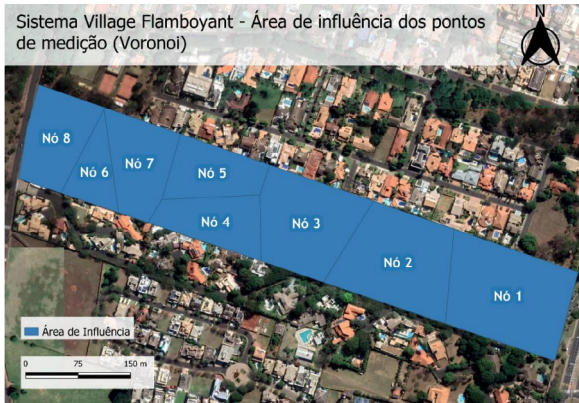
```

estimates = np.zeros((n, h))
residuals = np.zeros((n, h))
for t in range(h):
    "ENTRADA DE CONTROLE"
    u = S[:, t] - S[:, t - 1] if t > 0 else np.zeros(n)
    "Predição"
    x_pred = F @ X + B @ u
    p_pred = F @ P @ F.T + Q
    "Atualização"
    residual = M[:, t] - H @ x_pred
    residual_normalized = np.abs(residual) / M[:, t]
    residual_covariance = H @ p_pred @ H.T + R
    K = p_pred @ H.T @ np.linalg.
    inv(residual_covariance)
    # Atualizando o estado (X) e a matriz de covariância
    (P) do sistema
    X = x_pred + K @ residual
    P = (np.eye(n) - K @ H) @ p_pred
    estimates[:, t] = X
    residuals[:, t] = residual_normalized
    residuals_matrix = np.array(residuals)
    # Criando um DataFrame com os resíduos
    data = {f'Hora {i}': residuals[:, i] for i in range(24)}
    df = pd.DataFrame(data)
    df.index = [f'r{i + 1}' for i in range(8)]
    # Salvando o DataFrame em um arquivo Excel
    df.to_excel('residual.xlsx', index=True,
    engine='openpyxl')

```

#### 4 RESULTADOS

Para permitir uma análise geoespacializada, foi selecionado o software livre de código aberto QGIS que possui uma boa quantidade de *plugins*, além de permitir uma integração fluida com *Python*. Assim, a tabela “residual.xlsx” e as coordenadas dos pontos foram importadas, e foi utilizada a ferramenta de geoprocessamento “Polígonos de Voronoi” para determinar a área de influência de cada ponto, conforme se verifica na Figura 9.



**Figura 9** – Mapa temático “Polígonos de Voronoi”.

Os valores residuais obtidos pela execução do algoritmo do filtro de Kalman foram atribuídos para cada um dos polígonos por meio de correlações entre a tabela importada e a camada do nó correspondente.

#### 4.1 Segmentação de dados e definição de limiares para análise

Para possibilitar a detecção de anomalias através do QGis, é essencial segmentar os dados residuais resultantes em categorias que facilitem a interpretação e identificação visual de possíveis vazamentos. Nesse sentido, os dados foram segmentados em cinco categorias distintas, cada uma representando um intervalo de valores definido com base em desvios-padrão a partir da média dos resíduos:

- Valores normais: abrange os dados até  $\mu+1\sigma$ . Engloba os valores mais comuns e serve como referência para a identificação de variações.
- Variações comuns: compreende o intervalo de  $\mu+1\sigma$  até  $\mu+1.5\sigma$ , indicando variações que não possuem impacto importante na análise.
- Valores incomuns: estende-se de  $\mu+1.5\sigma$  até  $\mu+2\sigma$  e refere-se à incidência de valores indicativos de variações perceptíveis na dinâmica do sistema, mas sujeitos a uma baixa distinção com eventos não-anômalos, tais como uma demanda pontual para enchimento de uma piscina, abastecimento de um carro-pipa, entre outros.

- Possivelmente anômalos: é a zona de alerta intermediária que vai de  $\mu+2\sigma$  até  $\mu+2.8\sigma$ , que indica a necessidade de atenção sobre a condição do sistema.
- Anômalos confirmados: representa os dados acima de  $\mu+2.8\sigma$ , sendo os valores que indicam com maior assertividade a incidência de um vazamento no sistema.

A segmentação foi definida com base em propriedades estatísticas de uma distribuição gaussiana e ajustada para refletir as características específicas dos dados em análise. A escolha desses limiares busca permitir uma distinção clara entre variações normais e potenciais anomalias, facilitando a tomada de decisões e intervenções no sistema de distribuição de água. De modo geral, as categorias construídas possibilitaram perceber a variação, demonstrando que os valores residuais para o cenário estabelecido neste trabalho não ultrapassam 0,18 com o sistema em operação sem perturbações, como no exemplo do cenário às 9 horas (Figura 10), que se manteve em todos os cenários restantes. Houve apenas um aparente *outlier* no cenário das 14 horas, momento no qual foi registrado um valor de 0,20 no nó 2 (Figura 10), fato isolado e sem nenhuma correlação direta com algum evento controlado por este experimento, sendo provável que tenha existido algum ruído no sinal do sensor desse nó.

O resultado da análise da simulação do cenário com vazamento se deu às 15 horas (Figura 11), com a abertura do terminal de limpeza, que demonstrou o impacto dessa perturbação em relação à condição normal simulada pelo EPANET. Além disso, foi possível perceber a atuação do filtro de Kalman no horário subsequente à simulação do vazamento (Figura 11), realizando as correções do ganho de Kalman ( $k$ ) e reduzindo os residuais a patamares menores de 0,10 mesmo após a perturbação.



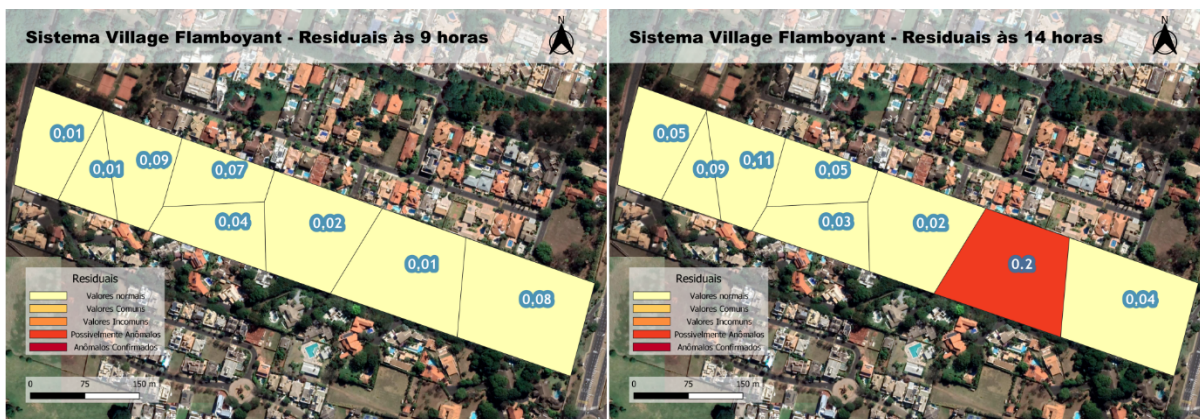


Figura 10 – Cenário às 9 horas (à esquerda); cenário às 14 horas e outlier no nó 2 (à direita).

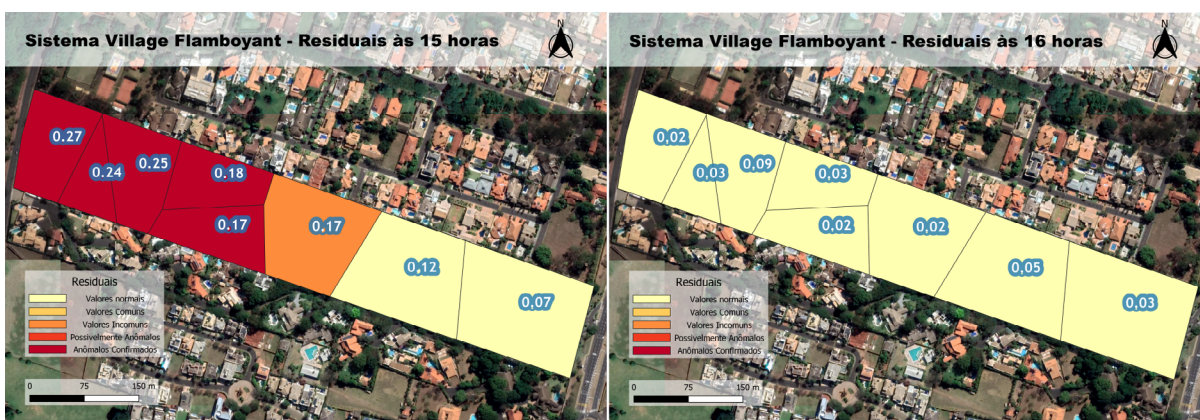


Figura 11 – Cenários às 15 horas, com vazamento simulado (à esquerda); cenário às 16 horas (à direita).

Outro ponto importante a se destacar é a delimitação correta da área com maior probabilidade de conter o vazamento. Tal condição pode permitir uma atuação rápida na intervenção dos setores operacionais e de manutenção, reduzindo tanto os valores gastos por vazamento localizado quanto o volume de água perdido até que seja corrigido.

## 5 DISCUSSÃO

A perspectiva deste trabalho pode reduzir os custos do programa de combate a perdas, virtualizando o conceito de Distrito de Medição e Controle (DMC)

e criando menores áreas de varredura. No entanto, há desafios técnicos a serem resolvidos, como a necessidade de energia local para os *dataloggers* e de infraestrutura de comunicação para transmitir dados para uma central de controle. Outros obstáculos incluem a necessidade de um cadastro confiável, bem como de sistemas setorizados, sem interligações não medidas e sem altos índices de perdas aparentes (fraudes e ligações clandestinas), eis que estes poderiam resultar em atribuição incorreta de demandas e, por conseguinte, em valores residuais inconstantes e sobrepostos aos valores de um possível vazamento.



## 6 CONCLUSÃO

Este estudo teve como objetivo desenvolver um modelo computacional aplicando o filtro de Kalman para detectar vazamentos em redes de distribuição de água. A partir dos resultados apresentados, verificou-se que a hipótese de utilização dos valores residuais como forma de controle ativo de anomalias nas redes de distribuição se confirmou de forma eficaz, demonstrando que a técnica é capaz de identificar perturbações de maneira verificável. A metodologia adotada se mostrou adequada ao problema apresentado pela pesquisa e permitiu a delimitação de potencialidades, tais como a virtualização do conceito de Distrito de Medição e Controle (DMC) e a criação de menores áreas de varredura. Entretanto, cabe ressaltar que desafios técnicos citados anteriormente, relacionados à infraestrutura necessária para a aplicação do método, ainda precisam ser superados em estudos futuros.

Os resultados obtidos são relevantes não apenas do ponto de vista acadêmico, mas também para gestores de recursos hídricos urbanos, uma vez que a abordagem pode contribuir para a redução de perdas e melhor gerenciamento de recursos. Por fim, é preciso salientar que o filtro de Kalman se mostrou uma ferramenta confiável para detectar perturbações em um sistema com operação normal. Contudo, sua característica principal é criar estimativas ponderadas entre os valores medidos e simulados; portanto, vazamentos inerentes, que não geram uma perturbação perceptível, ou vazamentos pré-existentes à aplicação do método podem ser incorporados pelo modelo sem gerar uma percepção clara do problema, sendo aconselhável que o sistema em que se dará a aplicação do método passe por um trabalho de redução de perdas prévio, com a utilização de métodos mais tradicionais, como o balanço hídrico e as vazões mínimas noturnas; posteriormente, o método descrito neste trabalho poderá ser utili-

zado como forma de monitoramento da condição das perdas reais do sistema.

## 7 CONTRIBUIÇÃO DOS AUTORES

**Conceitualização:** Zerunian RHM, Kellner E; **Metodologia:** Zerunian RHM, Kellner E; **Investigação:** Zerunian RHM, Kellner E; **Redação – Primeira versão:** Zerunian RHM, Kellner E; **Redação – Revisão & Edição:** Zerunian RHM, Kellner E; **Supervisão:** Zerunian RHM, Kellner E.

## 8 REFERÊNCIAS

- AL-WASHALI, T.; SHARMA, S.; KENNEDY, M. Methods of assessment of water losses in water supply systems: a review. **Water Resources Management**, Dordrecht, v. 30, p. 4985-5001, 2016.
- DELGADO-AGUIÑAGA, J.; BESANÇON, G.; BEGOVICH, O. Leak isolation based on Extended Kalman Filter in a plastic pipeline under temperature variations with real-data validation. *In: MEDITERRANEAN CONFERENCE ON CONTROL AND AUTOMATION (MED'15)*, 23., 2015, Torremolinos. **Proceedings** [...]. Torremolinos: Mediterranean Control Association, 2015.
- KIM, Y.; BANG, H. Introduction to Kalman Filter and its applications. *In: GOVAERS, F. (Ed.). Introduction and implementations of the Kalman Filter*. Londres: IntechOpen, 2019.
- KINGDOM, B.; LIEMBERGER, R.; MARIN, P. **The challenge of reducing non-revenue water (NWR) in developing countries** – how the private sector can help: a look at performance-based service contracting. Washington: The World Bank, 2006. (Water Supply and Sanitation Board Discussion, paper n. 8).
- PATHIRANA, A. **EpanetTools**: EPANET calling API for python. GitHub, 2015. Disponível em: <https://github.com/asselapathirana/epanettools>. Acesso em: 14 set. 2022.
- ROSSMAN, L. A. **EPANET 2: user's manual**. Cincinnati: National Risk Management Research Laboratory, 2000. Disponível em: <https://nepis.epa.gov/Adobe/PDF/P1007WWU.pdf>. Acesso em: 16 abr. 2024.
- RUZZA, V. **Data assimilation techniques for leakage detection in water distribution systems**. 2017. Tese (Doutorado em Ingegneria Civile ed Ambientale) – Dipartimento di Ingegneria Civile, Edile e Ambientale, Università Degli Studi di Padova, 2017.
- UNESCO; UN-WATER. **The United Nations world water development report 2020: water and climate change**. Paris: Unesco, 2020.