



**RODRIGO ANDRÉ FERREIRA**

**IMPLEMENTAÇÃO DE FERRAMENTA  
PARA AVALIAÇÃO DE DESEMPENHO  
DE MALHAS DE CONTROLE**

**LAVRAS – MG**

**2019**

**RODRIGO ANDRÉ FERREIRA MOREIRA**

**IMPLEMENTAÇÃO DE FERRAMENTA PARA AVALIAÇÃO  
DE DESEMPENHO DE MALHAS DE CONTROLE**

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Lavras, como parte das exigências do Curso de Engenharia de Controle e Automação, para a obtenção do título de Bacharel.

Prof. DSc. Vinicius Miranda Pacheco

Orientador

**LAVRAS – MG**

**2019**

**RODRIGO ANDRÉ FERREIRA MOREIRA**

# **IMPLEMENTAÇÃO DE FERRAMENTA PARA AVALIAÇÃO DE DESEMPENHO DE MALHAS DE CONTROLE**

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Lavras, como parte das exigências do Curso de Engenharia de Controle e Automação, para a obtenção do título de Bacharel.

APROVADO em 13 de junho de 2019.

DSc. Vinicius Miranda Pacheco UFLA

DSc. Fábio Domingues De Jesus UFLA

Giovanna Toledo Borges UFLA

Prof. DSc. Vinicius Miranda Pacheco

Orientador

**LAVRAS – MG**

**2019**

## RESUMO

Malhas de controle são parte fundamental para a execução de processos industriais e estão presentes em diversos ramos industriais. As malhas são robustas e projetadas para operar com um mesmo padrão de qualidade por tempo indeterminado. No entanto, elas foram projetadas para determinadas condições, e não é possível garantir que essas condições serão mantidas também por tempo indeterminado. Faz-se necessário uma forma de avaliar o desempenho dessas malhas de forma a identificar eventuais problemas e propor correções. O objetivo desse trabalho foi identificar, por meio de revisão bibliográfica e ferramentas disponíveis no mercado, índices para avaliar o desempenho de malhas de controle de forma escalável e automática. E posteriormente, seguindo uma metodologia que consistiu em definir as métricas a serem utilizadas, validar as fontes de dados utilizadas, e implementar algoritmos computacionais para calcular essas métricas. Foi possível implementar uma ferramenta disponível via *web* para apresentação das métricas calculadas e de uma forma que é possível estendê-las para outras malhas de controle de interesse. Essa ferramenta mostrou-se satisfatória para obter um diagnóstico preliminar sobre a malha de controle avaliada e também é passível de melhorias, como implementação de *benchmarking*.

**Palavras-chave:** Malhas de controle. PID. Índices de desempenho.

## **ABSTRACT**

Control loops are pivotal for the execution of industrial processes and can be found on every kind of industrial segment. The loops are robust and they are designed to work with a certain level of quality for a very long time. However, they are designed within certain conditions, and it is not possible to guarantee that these conditions will remain for a very long time. As such, it is necessary to evaluate the performance of these control loops in order to identify problems that may appear and propose corrections. The purpose of this work was to identify, through bibliographic revision and tools available in the market, scalable and automatic indexes to evaluate the performance of control loops. Subsequently, following a methodology that consisted in defining the metrics that will be calculated, validate the data sources that will be used, and implement computational algorithms to calculate these metrics. Eventually, a tool available through web was implemented that presented the calculated metrics and, in a way, that it is possible to extend them for other control loops. This tool was deemed satisfactory for obtaining a preliminary diagnostic about the evaluated control loop and subject to improvements, such as benchmarking.

**Keywords:** Control loops. PID. Performance indexes.

## SUMÁRIO

1	INTRODUÇÃO .....	7
2	REFERENCIAL TEÓRICO .....	9
2.1	Controladores PIDs .....	9
2.1.1	Termo Proporcional .....	9
2.1.2	Termo Integral .....	10
2.1.3	Termo Derivativo .....	11
2.2	Controlador Lógico Programável .....	12
2.3	Protocolo OPC .....	12
2.4	Índices de desempenho de PIDs .....	13
2.4.1	Detecção de Oscilação .....	13
2.4.2	<i>Idle Index</i> .....	14
2.4.3	Classificação da Oscilação .....	15
2.4.4	Mudanças de modo .....	16
2.4.5	Tempo em manual .....	17
2.4.6	Saturação do controle .....	17
2.4.7	Índice de Harris .....	17
2.5	Pandas .....	18
2.6	Amazon Web Services .....	18
2.6.1	Elastic Cloud Computing .....	19
3	METODOLOGIA .....	20
3.1	Escolha das malhas .....	20
3.2	Aquisição dos dados .....	20
3.3	Validação dos dados .....	21
3.4	Implementação das métricas .....	21
3.5	Apresentação dos resultados .....	22
4	RESULTADOS E DISCUSSÕES .....	24
4.1	Escolha da malha .....	24
4.2	Validação dos dados .....	24
4.2.1	Setpoint, Setpoint máximo e Setpoint mínimo .....	26
4.2.2	Variável Manipulada .....	28
4.2.3	Variável Controlada .....	28
4.2.4	Variável de status de funcionamento da malha .....	29
4.2.5	Variável de status automático/manual .....	30
4.2.6	Variáveis de status de funcionamento do sensor .....	30
4.2.7	Variável de status de funcionamento da correia de alimentação .....	32

<b>4.3</b>	<b>Implementação das métricas</b> .....	<b>32</b>
<b>4.3.1</b>	<b>Filtro de operação</b> .....	<b>32</b>
<b>4.3.2</b>	<b>Mudanças de modo</b> .....	<b>33</b>
<b>4.3.3</b>	<b>Tempo em manual</b> .....	<b>34</b>
<b>4.3.4</b>	<b>Saturação do atuador</b> .....	<b>35</b>
<b>4.3.5</b>	<b>Saturação do Controle</b> .....	<b>36</b>
<b>4.3.6</b>	<b>Erro médio</b> .....	<b>38</b>
<b>4.3.7</b>	<b>Tempo de operação da área</b> .....	<b>38</b>
<b>4.3.8</b>	<b>Tempo de operação do sensor</b> .....	<b>39</b>
<b>4.3.9</b>	<b>Normalização das métricas de tempo</b> .....	<b>39</b>
<b>4.4</b>	<b>Apresentação dos resultados</b> .....	<b>40</b>
<b>5</b>	<b>CONCLUSÃO</b> .....	<b>43</b>
<b>6</b>	<b>BIBLIOGRAFIA</b> .....	<b>45</b>

## 1 INTRODUÇÃO

O termo Indústria 4.0 é algo corrente na sociedade atual e reflete mudanças que estão acontecendo na indústria, e juntamente, trazendo uma grande mudança nos processos industriais. Indústria 4.0 é o nome para o novo padrão de automação e troca de dados nas tecnologias de processos industriais. São novos conceitos e premissas advindos da quarta revolução industrial, e que devido à grande profundidade dessas mudanças, e a velocidade com que elas estão sendo difundidas, traz uma quebra de paradigmas presentes nas indústrias. Isso exige que elas repensem seus modelos de funcionamento para se adequarem a essas mudanças.

Dentre as mudanças que compõem as “fábricas inteligentes”, estão inclusos sistemas ciberfísicos, internet das coisas e computação em nuvem. Esse padrão se baseia em seis pilares:

- Interoperabilidade;
- Virtualização;
- Descentralização;
- Capacidade em tempo-real;
- Orientação a serviço;
- Modularidade.

Esses pilares englobam justamente a capacidade de coletar e analisar dados, e entregar conhecimento derivado dessas análises. Análises que serão de grande importância na tomada de decisões sobre o processo, de forma a obter diversas melhorias como otimização do uso de matéria prima, redução do uso de combustíveis, maior qualidade do produto final, etc.

É necessário também destacar a importância da computação em nuvem, que permite o armazenamento desse grande volume de dados em um servidor sem a necessidade de investir numa infraestrutura dedicada a essa tarefa. Além disso, esses dados também podem ser acessados de qualquer lugar por meio da internet, descentralizando a informação e permitindo que essas análises possam ser estudadas e, conseqüentemente, decisões sejam tomadas também de forma descentralizada.

Quando se trata de análise de processos em uma indústria, diversos são os fatores que podem influenciar o processo como um todo, tais como a taxa de consumo de energia elétrica ou de combustível por turno de operação, taxa de descarte do produto final, entre outros aspectos mais específicos de cada processo, sendo possível inclusive desmembrar cada fator em diversos subfatores. Tendo em vista tamanho leque de possibilidades é possível que a pessoa responsável



opte por fazer todas as análises possíveis de uma vez. No entanto, essa não é a estratégia mais viável visto que seria uma tarefa muito dispendiosa, e uma vez que fosse concluída, tais análises poderiam se mostrar desnecessárias. Uma forma mais branda de implementar tal análise, seria de forma gradual, iniciando-se por algo relativamente mais simples, mas que seria impactante no processo como um todo, como malhas de controle PID.

Malhas de controle PID são consideradas robustas, e uma vez projetadas raramente sofrem alterações. Essas malhas de fato funcionam de forma adequada por tempo indeterminado, desde que as condições com que elas foram projetadas, sejam mantidas. No entanto, essa premissa não é verdadeira no ambiente industrial. Existem inúmeras mudanças que podem ocorrer que influenciarão a malha de controle, como deterioração do atuador ou do sensor, mudanças na planta, adição de distúrbios, etc. Um aspecto que recebe bastante destaque é a oscilação nas malhas. Oscilações em uma malha de controle podem se propagar de uma malha para outra, caso estejam acopladas, acarretando em um alto consumo de energia, desperdício de matéria prima e produto final não uniforme [Hägglund, 1995].

Logo, faz-se necessário uma análise contínua dessas malhas de PID de forma a garantir que o desempenho das mesmas está adequado. Esse tipo de análise é algo simples, mas que demanda algum tempo para ser implementado. No entanto, quando se pensa no imenso volume de dados produzido por uma indústria diariamente, e no número de malhas de PID presentes em uma planta, essa análise é deixada de lado. Para remediar essa situação, a melhor estratégia seria a implementação de um algoritmo que avalie o desempenho de malhas de PID de forma genérica, possibilitando a sua replicação para todas as malhas existentes em uma planta, e garantindo também uma extensibilidade para eventuais ampliações que venham a ser implementadas.

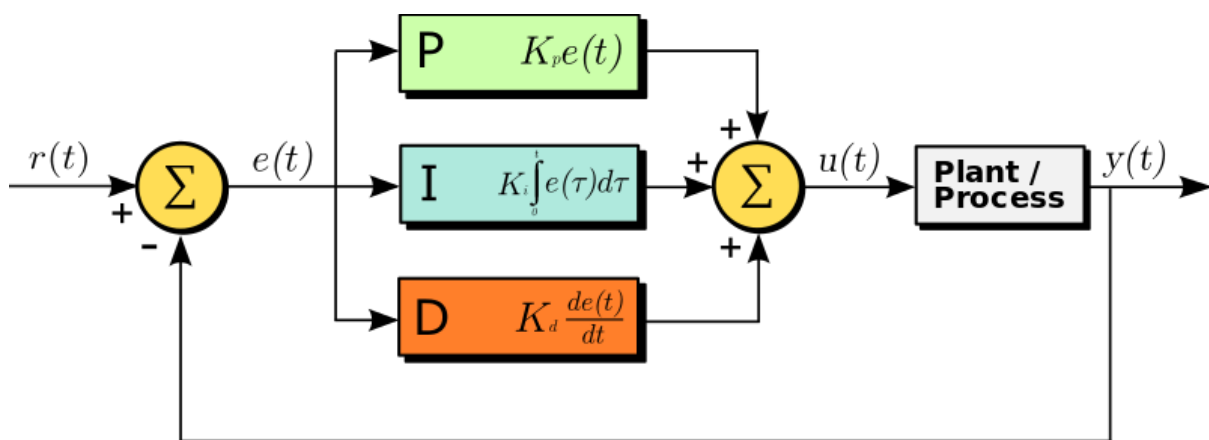
O objetivo deste trabalho é investigar uma forma de avaliar malhas de controle de forma automática e escalável, para identificar a necessidade de manutenção, seja numa nova sintonia do controlador, manutenção do atuador ou do instrumento de medição, e evitar possíveis deteriorações desnecessárias ao atuador causadas por oscilações provenientes da malha de controle e outros fatores. E posteriormente, implementar uma ferramenta para calcular esses índices e apresentar os resultados de forma visual e intuitiva.

## 2 REFERENCIAL TEÓRICO

### 2.1 Controladores PIDs

Controlador PID (Proporcional-integral-derivativo) é um mecanismo de controle em malha fechada amplamente utilizado em sistemas de controle industriais entre outras aplicações que necessitam de um controle contínuo. Um controlador PID calcula de forma contínua o valor de erro  $e(t)$ , que é a diferença entre o setpoint (SP) e a variável de processo (PV). Em seguida ele aplica uma correção baseada em nos três termos: proporcional, integral e derivativo, na forma de um sinal que é a saída do controlador PID,  $u(t)$ .

Figura 1: Diagrama de blocos de um controlador PID em série



Fonte: Arturo Urquizo (2011)

O ajuste fino desses três termos resulta na obtenção de parâmetros de desempenho desejados para o sistema de controle, como:

- Overshoot (máximo sobressinal): o máximo valor que excede o setpoint.
- Tempo de subida: tempo necessário para o sinal ir de 10% para 90% do valor final.
- Tempo de assentamento: tempo necessário para que o sinal permaneça em uma faixa de valores de 2% em torno do valor final.

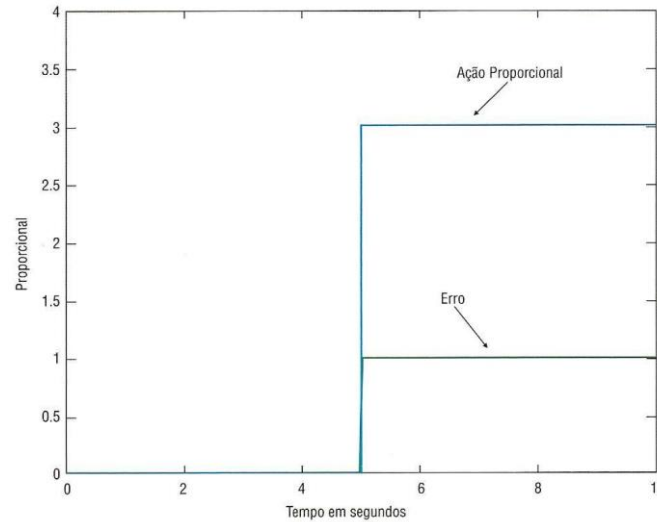
Cada termo produz um sinal de saída distinto em função do erro, de forma a eliminar esse desvio.

#### 2.1.1 Termo Proporcional

Esse termo gera uma saída proporcional ao erro. É um fator multiplicativo também conhecido como o ganho do controlador. Esse fator deve ser escolhido corretamente em função

do processo, uma vez que a escolha desse fator pode desestabilizar o sistema. Além disso, o uso de apenas o termo proporcional não garante a eliminação do erro em regime permanente.

Figura 2: Ação proporcional



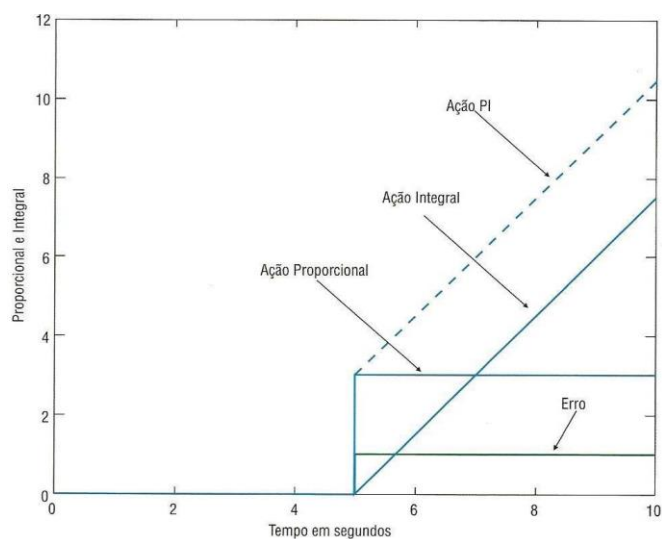
Fonte: de Campos (2010)

### 2.1.2 Termo Integral

A ação integral gera uma saída que é a integral do erro. Esse sinal irá crescer ou decrescer indefinidamente enquanto existir um sinal de erro. Por esse motivo, a ação integral elimina o erro em regime permanente, no entanto isso pode provocar a saturação do atuador. Caso esse comportamento não seja considerado, é possível que o sinal de controle alcance valores acima da saturação do atuador, por exemplo 120%, e quando ocorrer uma necessidade de que o atuador opere em valores menores, será necessário aguardar que esse sinal alcance valores abaixo de 100% para que ocorra alguma alteração no estado do atuador.

O termo integral é composto pelo fator multiplicativo  $1/T_I$ , onde  $T_I$  é o tempo integral, geralmente em segundos ou minutos por repetição. Esse tempo representa o tempo necessário para a ação integral “repetir” a ação proporcional.

Figura 3: Ação integral e proporcional



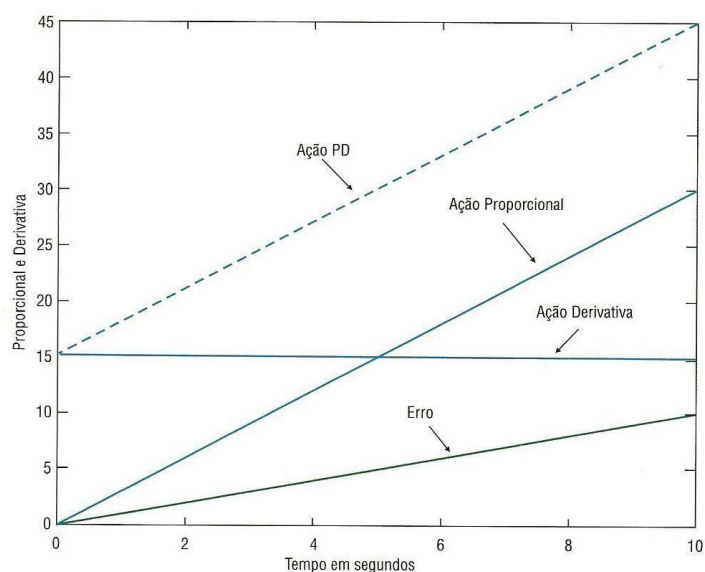
Fonte: de Campos (2010)

### 2.1.3 Termo Derivativo

De forma análoga, a ação derivativa gera uma saída que é proporcional à derivada do erro. Esse termo tenta estimar a tendência do comportamento do sinal do erro.

A ação derivativa é composta pelo fator multiplicativo TD, que é o tempo derivativo. De forma análoga ao tempo integral, o tempo derivativo representa o tempo que a ação derivativa antecipa a ação proporcional. Ou seja, após um tempo TD, a ação proporcional será igual à ação derivativa.

Figura 4: Ação derivativa e ação proporcional



Fonte: de Campos (2010)

## **2.2 Controlador Lógico Programável**

Controladores lógicos programáveis (CLPs) são computadores digitais industriais que foram adaptados e tornados robustos para desempenharem o papel de controlar processos industriais.

Esses computadores foram desenvolvidos com intuito de substituir sistemas de controle baseados em relés, que apesar de serem eficazes para desempenhar esse papel, demandavam um grande esforço para serem projetados, uma vez que cada sistema pode conter de dezenas a milhares de relés dependendo da complexidade do tipo de controle. Outro ponto negativo dos sistemas de relés é proveniente da dificuldade de expansão e manutenção, uma vez que o sistema consiste de, além dos relés propriamente ditos, ligações elétricas, o que acarreta em possíveis falhas tanto mecânicas advindas das bobinas dos relés quanto de falhas elétricas provenientes dos cabos utilizados para as ligações elétricas.

Os CLPs foram criados com o intuito de substituir as conexões físicas entre os relés por conexões digitais, ou seja, a lógica de abertura e fechamento de contato dos relés seria executada digitalmente via programação, de forma que não seria mais necessário dar manutenção a uma infinidade de relés e ligações elétricas entre eles.

O princípio de funcionamento dos CLPs é baseado num ciclo de tarefas, primeiramente são lidas as entradas, que geralmente são sinais de corrente provenientes de sensores disponíveis na planta; em seguida é executada a rotina que foi implementada, essa rotina pode ser por exemplo, o cálculo da saída de um controlador PID, que será enviado para um atuador; e por último são atualizadas as saídas, utilizando o último exemplo, é a tarefa responsável por enviar o sinal de controle calculado para o atuador.

## **2.3 Protocolo OPC**

O protocolo OPC (Open Platform Communications) é constituído de uma série de padrões e especificações para comunicação industrial voltada para o segmento da automação. Ele foi desenvolvido para garantir a comunicação de softwares e hardwares de controle de processos.

Cada fabricante de CLP possui seus próprios padrões para implementação e execução de rotinas. Esses padrões seguem o mesmo conceito, mas diferem na execução. Uma boa analogia seria o conceito de idioma, onde cada fabricante ensina seu idioma para seu CLP, e apesar dos idiomas terem o mesmo objetivo, não necessariamente um CLP de determinado fabricante conseguirá se comunicar com um CLP de um outro fabricante. O protocolo OPC funciona como

um segundo idioma que os CLPs podem aprender, garantindo que eles consigam comunicar entre si e qualquer outro dispositivo ou software que conheça esse idioma.

Esse protocolo é de grande importância para aquisição de dados provenientes de CLPs, pois uma vez que o software responsável por essa aquisição consegue se comunicar com qualquer CLP que possua compatibilidade com esse protocolo, garantindo que a aquisição de dados seja feita independente dos modelos de CLPs presentes na planta.

No mercado, existem diversas ferramentas responsáveis por realizar essa comunicação, como Kepware, Matrikon, etc. São *softwares* que fazem o papel de servidor OPC, possuindo os módulos para estabelecer a comunicação com os diversos CLPs existentes no mercado, e o papel de repassar essas informações obtidas pela comunicação para os clientes OPC que possam comunicar com a máquina onde esse servidor está instalado.

## 2.4 Índices de desempenho de PIDs

Existem diversos índices que avaliam o desempenho de uma malha de controle PID. Neste trabalho serão utilizados índices que avaliam a presença e diagnóstico de oscilações em malhas de controle, além de outras métricas que avaliam outros fatores como dimensionamento e a própria utilização da malha.

### 2.4.1 Detecção de Oscilação

Como foi mencionado anteriormente, oscilações são prejudiciais ao processo, e caso existam oscilações nas malhas de controle, é necessário realizar um diagnóstico de forma a removê-las. Primeiramente é necessária uma estratégia para detectar a existência de oscilações na malha de controle para que então sejam feitas análises referentes a causa da mesma (Hägglund, 1995). Esse procedimento é baseado na análise da integral absoluta do erro (IAE) entre cruzamentos do eixo x, segundo a equação 1.

$$IAE = \int_{t_{i-1}}^{t_i} |e(t)| dt \quad (1)$$

Onde  $t_{i-1}$  e  $t_i$  são duas incidências consecutivas de cruzamentos do eixo x pelo erro, considerando que a média do sinal de erro do controle é zero. Quando o controle está satisfatório, a magnitude do erro do controle é pequena, e a o tempo entre os cruzamentos do eixo x são curtos. Em contrapartida, quando ocorre um distúrbio de carregamento, a magnitude do erro aumenta, e o tempo entre cruzamentos aumenta. Consequentemente, o índice IAE aumenta. Ao comparar esse valor de IAE com um valor limite  $IAE_{lim}$ , pode se concluir que

ocorreu um distúrbio de carregamento, e esse limite pode ser obtido utilizando a frequência máxima do sistema segundo a equação 2.

$$IAE_{lim} = \frac{2}{\omega_u} \quad (2)$$

Onde  $\omega_u$  é a frequência máxima do sistema, que pode ser obtida a partir do tempo integral ( $T_i$ ) de acordo com a equação 3.

$$\omega_u = \frac{2\pi}{T_i} \quad (3)$$

A partir da detecção desses distúrbios de carregamento, pode-se considerar que ocorre uma oscilação quando a frequência desses distúrbios fica alta. Se durante um período  $T_{sup}$ , o número de detecções for maior que um limite  $n_{lim}$ , conclui-se que existe uma oscilação. O período de supervisão,  $T_{sup}$ , pode ser calculado através da relação apresentada na equação 4.

$$T_{sup} = 5 \times n_{lim} \times T_i \quad (4)$$

Uma vez detectada a oscilação através desse algoritmo, o próximo passo é diagnosticar a oscilação: determinar se a oscilação é gerada dentro ou fora da malha, encontrar a fonte geradora da oscilação e corrigir o problema.

#### **2.4.2 Idle Index**

Assim como oscilações são prejudiciais ao processo, controladores mal sintonizados causam perdas tanto na produção, quanto na qualidade do produto final (Bialkowski, 1993; Ender, 1993). Muitas malhas de controle de processos industriais são sintonizadas de forma conservativa. Apesar dessa característica não provocar oscilações ou overshoots, esses controladores tendem a responder de forma lenta quando ocorrem distúrbios de carregamento, acarretando em grandes e longos desvios desnecessários em relação ao set point, e consequentemente, em uma redução da qualidade do produto final.

É necessário também um índice que indique o quão lenta é a resposta do controlador, caracterizando o controlador como conservativo. *Idle index*, que parte do princípio que um controlador está bem sintonizado quando ele apresenta uma resposta rápida quando ocorrem distúrbios de carregamento, mas sem overshoot. Esse índice analisa a correlação entre as variações do sinal de controle e o sinal de saída (Hägglund, 1999).

Primeiramente são calculados os períodos de tempo em que as correlações entre os incrementos dos sinais são positivas e negativas, respectivamente segundo as equações 5 e 6.

$$t_{pos} = \begin{cases} t_{pos} + h, & \Delta u \Delta y > 0 \\ t_{pos}, & \Delta u \Delta y \leq 0 \end{cases} \quad (5)$$

$$t_{neg} = \begin{cases} t_{neg} + h, & \Delta u \Delta y < 0 \\ t_{neg}, & \Delta u \Delta y \geq 0 \end{cases} \quad (6)$$

Onde  $h$  é o período de amostragem.

O *Idle Index*  $I_i$ , é definido então segundo a equação 7.

$$I_i = \frac{t_{pos} - t_{neg}}{t_{pos} + t_{neg}} \quad (7)$$

Esse índice varia de -1 até 1, onde um valor positivo e próximo a 1 indica um controle lento. Enquanto um valor negativo próximo de -1, indica um controlador bem sintonizado. No entanto, valores negativos também são obtidos a partir de malhas oscilatórias, logo, é interessante utilizar o *Idle Index* juntamente com um procedimento para detecção de oscilações. Valores pequenos, próximos de 0 ( $-0.4 < I_i < 0.4$ ), também indicam uma malha de controle bem sintonizada.

### 2.4.3 Classificação da Oscilação

Uma vez detectada, é necessário classificar o que está provocando a oscilação na malha de controle. Um método para atingir esse objetivo através do cálculo da função de correlação cruzada entra o sinal de saída  $y(t)$  e o sinal de controle  $u(t)$  (*Horch, 1999*), através da equação 8.

$$r_{uy}(\tau) = \sum_{k=0}^{N-|\tau|-1} u(k)y(k + \tau) \quad (8)$$

Uma vez calculada a função de correlação cruzada, classifica-se essa função em par ou ímpar. Para determinação automática, é necessária definir algumas variáveis e as relações expostas pelas equações 9, 10 e 11.

$\tau_r =$  cruzamento com eixo  $x$  para atrasos positivos

$-\tau_l =$  cruzamento com eixo  $x$  para atrasos negativos

$r_0 =$  correlação cruzada no lag 0

$$r_{max} = \text{sign}(r_0) * \max_{\tau \in [-\tau_l, \tau_r]} |r_{uy}(\tau)| \quad (9)$$

$$\Delta\tau = \frac{|\tau_l - \tau_r|}{|\tau_l + \tau_r|} \quad (10)$$



$$\Delta Q = \frac{|r_0 - r_{max}|}{|r_0 + r_{max}|} \quad (11)$$

As variáveis  $\Delta\tau$  e  $\Delta Q$  serão então utilizadas para determinar o atraso  $\Delta\varphi$  entre os sinais  $u(t)$  e  $y(t)$ . Como o objetivo é distinguir esse atraso entre  $\pi/2$  (função de correlação cruzada ímpar) e  $\pi$  (função de correlação cruzada par), é possível utilizar as seguintes aproximações:

$$\Delta\varphi = \pi/2 \pm \pi/6 = \pi/2$$

$$\Delta\varphi = \pi \pm \pi/6 = \pi$$

Para valores fora desse intervalo, no caso:  $\Delta\varphi \in [\pi/2 + \pi/6, \pi - \pi/6]$ , nenhum diagnóstico será dado de modo a evitar indicações errôneas. Utilizando os valores de  $\Delta\tau$  e  $\Delta Q$ , podem ser traduzidos da seguinte forma:

$$\Delta\varphi = \pi \leftarrow \begin{cases} 0.0 < \Delta Q \leq \frac{2-\sqrt{3}}{2+\sqrt{3}} \\ 0.0 < \Delta\tau \leq \frac{1}{3} \end{cases} \quad (12)$$

$$\text{Nenhuma decisão} \leftarrow \begin{cases} \frac{2-\sqrt{3}}{2+\sqrt{3}} < \Delta Q \leq \frac{1}{3} \\ \frac{1}{3} < \Delta\tau \leq \frac{2}{3} \end{cases} \quad (13)$$

$$\Delta\varphi = \frac{\pi}{2} \leftarrow \begin{cases} \frac{1}{3} < \Delta Q < 1.0 \\ \frac{2}{3} < \Delta\tau < 1.0 \end{cases} \quad (14)$$

Uma vez determinado o valor de  $\Delta\varphi$ , a função de correlação cruzada pode ser caracterizada sendo par para  $\Delta\varphi = \pi$ , e ímpar para  $\Delta\varphi = \pi/2$ . Essa caracterização irá então indicar o motivo da oscilação. Tipicamente, essa função é par quando distúrbios oscilantes externos entram na malha de controle. Em contrapartida, a função é ímpar quando o distúrbio oscilante é interno à malha de controle, como problemas de uma válvula que esteja agarrando, resultando em um comportamento de histerese.

#### 2.4.4 Mudanças de modo

Dentre os benefícios da utilização de malhas de controle, está a redução da necessidade de intervenção do operador, tanto para aumentar a uniformidade do produto, quanto para permitir que o operador desempenhe outras funções ao mesmo tempo. Geralmente, o operador precisa fazer algum tipo de intervenção quando ocorre algum distúrbio na malha de controle caso a própria malha não seja capaz de assimilar essa ocorrência. Uma malha que o operador precisa

intervir de forma frequente não está trazendo o benefício ao qual ela foi proposta, indicando que alguma decisão precisa ser tomada para corrigir esse problema, seja conduzindo uma nova sintonia do controlador, seja substituindo atuadores ou sensores, ou outra alternativa.

#### 2.4.5 Tempo em manual

De forma análoga à métrica anterior, o número de mudanças no modo de operação da malha não é suficiente para avaliar o grau da intervenção do operador. Uma frequência alta de mudanças de modo de operação indica que o operador precisa intervir no processo frequentemente, no entanto não é possível, por exemplo, concluir se essas intervenções foram dispendiosas ou não, pois não se sabe por quanto tempo a malha ficou em manual. Faz-se necessário avaliar também quanto tempo a malha trabalhou em modo manual/automático de forma a quantificar quão independente ela está operando, ou quanto tempo durou a intervenção do operador.

#### 2.4.6 Saturação do controle

Malhas de controle são projetadas para operar numa escala mediana em relação aos limites especificados. Projeta-se uma malha levando em consideração o intervalo de operação do processo, utilizando atuadores e sensores que operem numa faixa que inclua esse intervalo, mas de forma que esse intervalo não esteja nos limites de operação. No entanto, é possível que a malha esteja operando nas extremidades, e esse tipo de operação pode acarretar num desgaste prematuro do atuador, sendo necessária uma reavaliação do dimensionamento do mesmo ou do próprio controlador.

#### 2.4.7 Índice de Harris

O índice de Harris é o resultado da comparação entre a variância do sinal de saída do sistema atual  $\sigma_y^2$  com a variância do sinal de saída obtida por um controlador de variância mínima  $\sigma_{MV}^2$  aplicado a um modelo de série temporal obtido com auxílio dos dados de saída.

O índice de Harris (equação 15) é resultado da seguinte relação entre a variância mínima do sistema ( $\sigma_{MV}^2$ ) e a variância real do sistema ( $\sigma_y^2$ ).

$$\eta_{MV} = \frac{\sigma_{MV}^2}{\sigma_y^2} \quad (15)$$

Onde:

$$\sigma_{MV}^2 = \sum_{i=0}^{\tau-1} e_i^2 \sigma_\varepsilon^2 \quad (16)$$

$$\sigma_y^2 = \sum_{i=0}^{\infty} e_i^2 \sigma_\varepsilon^2 \quad (17)$$

E o valor desse índice varia entre 0 e 1, onde valores próximos de 1 indicam um bom controle com respeito a variância de saída obtida pelo modelo teórico, enquanto 0 indica uma performance pior, incluindo controle instável.

Esse índice pode ser calculado independentemente do tipo de controlador desde que as seguintes informações sejam conhecidas:

- Dados coletados referentes à variável controlada.
- Valor real ou estimação do tempo morto do sistema ( $\tau$ ).

A partir desses dados é possível modelar o sistema em malha fechada utilizando as informações coletadas e obter uma função de transferência. Uma vez obtida, é possível estimar a variância mínima dessa malha de controle ( $\sigma_{MV}^2$ ) e compará-la com a variância real do sistema ( $\sigma_y^2$ ).

## 2.5 Pandas

*Pandas* é uma biblioteca escrita em linguagem *Python* utilizada para análise de dados. A linguagem *Python* é considerada ótima para preparação de dados para análise, no entanto, quando se trata de análise e modelagem dos dados ela não possui tantas funcionalidades. A biblioteca *pandas* ajuda a suprir essa necessidade, possibilitando essa etapa de análise de dados em *Python*, ao invés de quebrar esse processo de trabalho por ter que utilizar uma outra ferramenta mais apropriada, como *R*.

Dentre as funcionalidades presentes nessa biblioteca, é possível destacar:

- Objetos do tipo *DataFrame* para manipulação de dados com índice integrado;
- Alinhamento de dados e ferramentas para tratar dados faltantes;
- Inserção e remoção de colunas em estruturas de dados;
- Ferramentas para combinação e agrupamentos em conjuntos de dados;
- Geração de intervalos de datas, conversão de frequências, estatísticas de janelas rolantes, etc.

## 2.6 Amazon Web Services

Amazon Web Services (AWS) é uma plataforma segura de serviços de computação em nuvem, que oferece serviços de armazenamento, poder computacional, entrega de conteúdo e outras funcionalidades de forma prática e escalável. A AWS conta com mais de 55 serviços e

disponíveis em 21 regiões diferentes, incluindo São Paulo. Para esse projeto foi utilizado apenas o Elastic Cloud Computing.

### **2.6.1 Elastic Cloud Computing**

O Elastic Cloud Computing (EC2) é um *web service* que oferece capacidade computacional escalável e segura na nuvem. É tecnicamente uma máquina (instância) que é possível ter total acesso aos recursos computacionais, permitindo também o dimensionamento da capacidade rapidamente à medida que os requisitos mudem. A instância utilizada nesse projeto foi responsável por receber, armazenar e controlar o acesso aos dados provenientes da planta, e também por realizar as análises utilizando os mesmos.

### **3 METODOLOGIA**

#### **3.1 Escolha das malhas**

Como foi mencionado inicialmente, uma planta possui inúmeras malhas de controle em sua cadeia de processos. Inicialmente é interessante escolher as malhas de maior impacto para o processo, maior custo operacional ou algum outro motivo de interesse.

Para esse estudo, a malha avaliada foi escolhida por se tratar de uma malha que é utilizada para um controle especialista, ou seja, é uma malha de controle que é utilizada para implementar uma estratégia de controle mais avançada. É uma malha de interesse pois para que essa estratégia de controle tenha sucesso, é necessário garantir que ela esteja operando adequadamente.

É importante ressaltar que como primeiro passo é interessante escolher as malhas de maior impacto de forma a dar maior ênfase em analisar e otimizá-las, no entanto, pelo caráter expansível dessas análises e métricas, também é interessante expandir de forma gradual para as outras malhas existentes.

#### **3.2 Aquisição dos dados**

Uma vez definidas as métricas a serem utilizadas e as malhas a serem avaliadas, faz-se necessário estabelecer a comunicação entre a planta de onde serão gerados os dados, e a nuvem, onde serão armazenados.

Os dados são provenientes de CLPs existentes na planta, e o envio dos dados será feito utilizando um computador também localizado na planta. Primeiramente é preciso estabelecer a comunicação entre o CLP e a máquina, e essa comunicação é feita com auxílio do protocolo OPC. Para o estudo de caso desse trabalho, a máquina operou tanto como cliente quanto como servidor para essa comunicação.

Em seguida, devem ser identificadas as variáveis a serem enviadas para o cálculo das métricas:

- Setpoint;
- Variável manipulada;
- Variável controlada;
- Variável de status automático/manual;
- Variável de status de funcionamento da malha;

- Variável de status de funcionamento do sensor (opcional, uma vez que a análise de funcionamento do sensor pode ser feita utilizando limites de medição).

De posse dos endereços dessas variáveis, o próximo passo é configurar o envio das leituras das mesmas. Para esse estudo, foi utilizado o *software* da empresa Optimus para realizar essa tarefa. O *software* é instalado na máquina cliente OPC, e a configuração do envio é feita via web, onde são definidos os endereços das variáveis a serem enviadas, o nome a ser utilizado como identificador das medições dessa variável no banco de dados e o período de amostragem.

### **3.3 Validação dos dados**

Enviados os dados, é importante garantir a consistência dos mesmos e que de fato os valores são referentes às variáveis associadas (garantir que não foi utilizado um endereço errado por exemplo). Para o caso desse trabalho, foi feito o uso do framework da empresa Optimus para acessar os dados enviados.

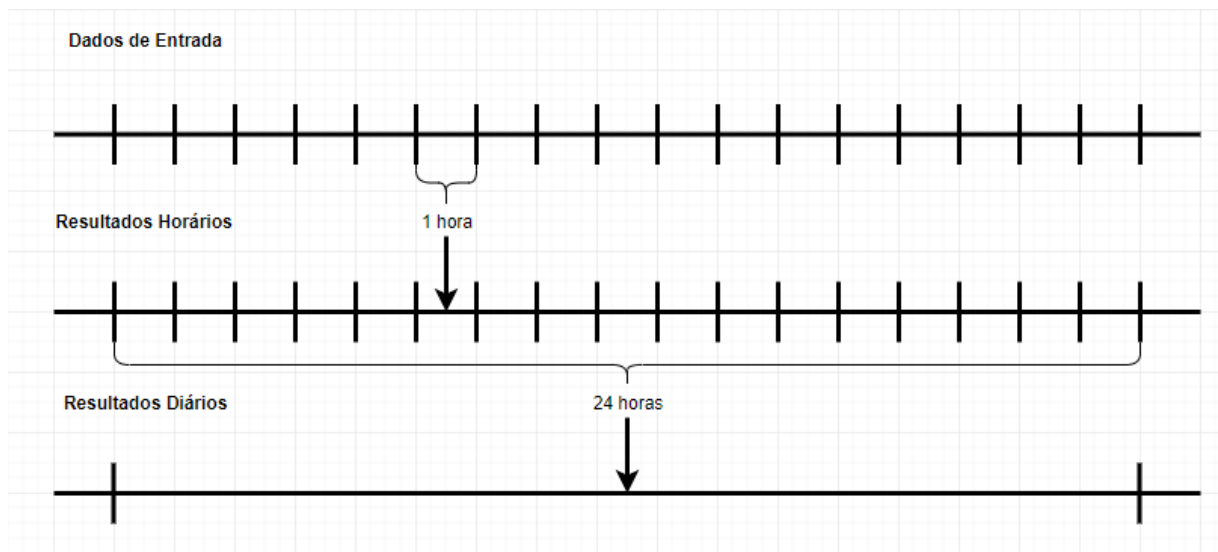
O framework opera utilizando a linguagem de programação *Python* e os dados podem ser operados fazendo uso das bibliotecas disponíveis (*Pandas*, *numpy*, etc.). Também é interessante avaliar se as variáveis de medições (manipulada, controlada) estão com valor percentual. É possível encontra-las em valor absoluto, valor de corrente, unidade de engenharia, etc., e caso isso ocorra, é necessário realizar uma conversão para possibilitar o cálculo das métricas.

### **3.4 Implementação das métricas**

Com os dados em mãos e validados, o próximo passo é implementar as rotinas a serem executadas para realizar os cálculos das métricas. O framework utilizado possibilita de forma análoga ao acesso dos dados armazenados, operar sobre esses dados de forma constante, e armazenar também os resultados dessas operações em banco de dados.

A estratégia utilizada para realizar os cálculos consiste em seccionar os dados de entrada em intervalos de uma hora, executar os cálculos que possam ser acumulados, como por exemplo totalizar o tempo em que a malha operou em modo manual, e armazenar esses resultados no banco de dados. Em seguida, os resultados horários serão utilizados para calcular o resultado referente ao dia analisado, e assim por diante.

Figura 5: Estratégia para realização dos cálculos



Fonte: do autor (2019)

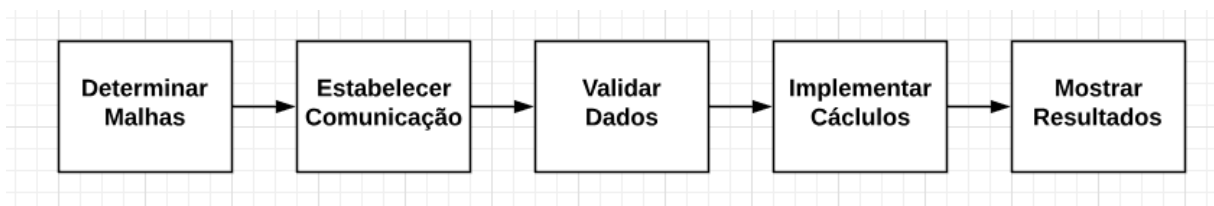
Cada resultado tem um valor calculado, um identificador referente ao que está sendo calculado (tempo em manual por exemplo), e um *timestamp* que representa a qual intervalo de tempo o valor calculado é referente.

### 3.5 Apresentação dos resultados

Uma vez calculadas as métricas, o último passo é apresentá-las de forma legível e de fácil entendimento. Foi utilizado o framework de *frontend* da empresa Optimus para apresentação desses resultados via web.

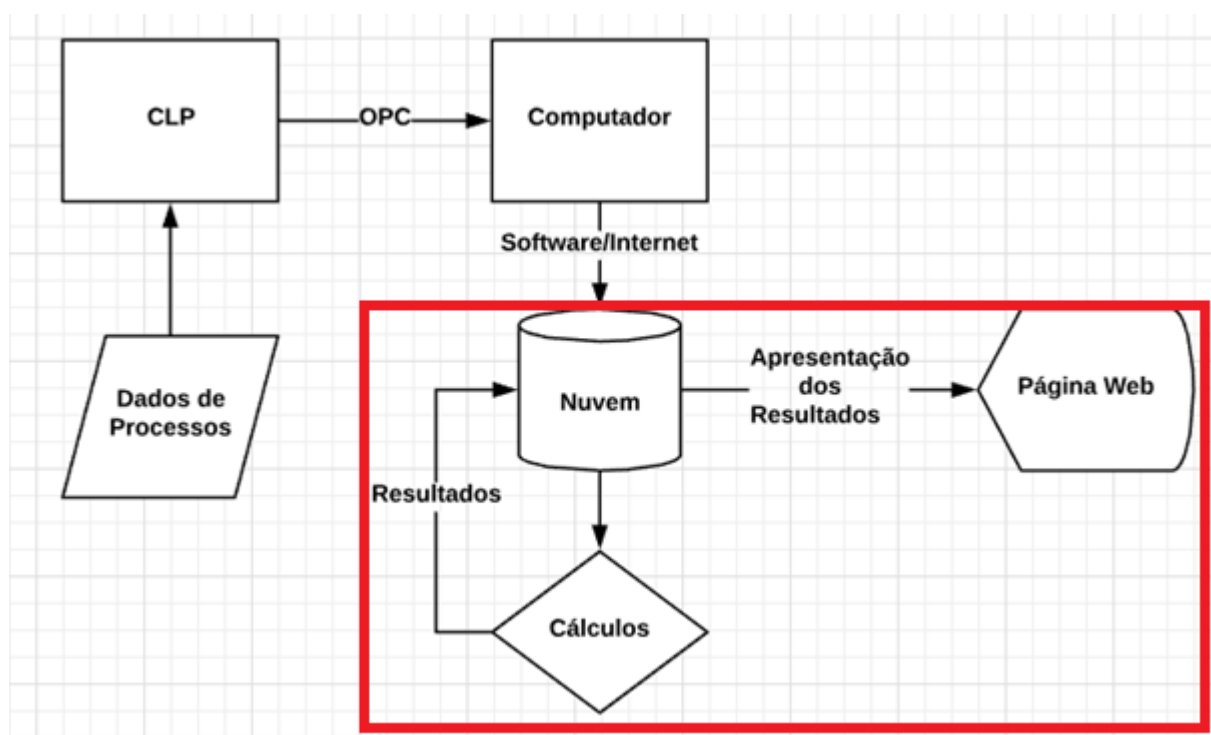
Foi adotado uma estratégia modular para a apresentação dos dados. A página inicial apresenta todas as malhas que estão sendo analisadas e as respectivas porcentagens de tempo em que cada malha operou em automático. Cada malha possui um botão para redirecionar à página que contém as métricas calculadas referentes àquela malha. É possível alterar a data de análise e também o intervalo de tempo de análise, seja dia, ou mensal.

Figura 6: Fluxograma da metodologia



Fonte: do autor (2019)

Figura 7: Fluxograma das estruturas



Fonte: do autor (2019)

É importante ressaltar que o foco deste trabalho está concentrado nos itens em destaque, especificamente a implementação dos cálculos a partir dos dados armazenados em nuvem segundo uma convenção, e a apresentação dos resultados em página web. Os outros objetos do fluxograma e respectivas interfaces serão tratados de forma superficial.



## 4 RESULTADOS E DISCUSSÕES

### 4.1 Escolha da malha

Conforme foi introduzido no tópico 3.1, a malha escolhida para esse estudo é referente a uma malha de controle da taxa de alimentação de um moinho de minério. Nessa malha, existe um inversor de frequência ligado a um motor responsável por controlar a velocidade de uma correia transportadora de minério. Também existe uma balança acoplada à correia, responsável por medir a vazão de minério. Por último, há também um CLP que atua como controlador dessa malha, variando a frequência do inversor para obter a taxa de alimentação desejada, cujo valor é calculado por um controle especialista baseado em lógica Fuzzy. Em resumo:

- *Setpoint*: *setpoint* para a taxa de toneladas de minério por hora cujo valor é obtido com auxílio do controle especialista à partir dos valores limites estipulados pelo operador;
- *Setpoint Mínimo e Máximo*: variáveis inseridas pelo operador para intervalo de operação da variável controlada;
- Variável manipulada: sinal enviado ao inversor de frequência responsável por controlar a velocidade da correia transportadora;
- Variável controlada: taxa de toneladas de minério por hora medido por uma balança presente na correia transportadora;
- Variável de status automático/manual: variável booleana proveniente da lógica implementada no CLP;
- Variável de status de funcionamento da malha: variável booleana proveniente da lógica implementada no CLP (controle atuando ou não);
- Variável de status de funcionamento do sensor: variáveis booleanas provenientes da lógica implementada no CLP para análise dos valores da variável controlada.

### 4.2 Validação dos dados

Uma vez cadastrados os endereços do CLP referente aos dados de interesse e os mesmos terem sido enviados, foi possível avaliar a consistência dos dados.

A figura 8 apresenta o *Dataframe* transposto (para facilitar a leitura) dos dados coletados referentes ao dia 23 de abril. Cada linha representa uma variável diferente, e cada coluna o momento em que o dado foi amostrado. O período de amostragem é de 15 segundos, logo cada

coluna será um timestamp no formato ano-mês-dia, hora:minuto:segundo + fuso horário, e colunas consecutivas terão uma diferença de 15 segundos em seu timestamp.

Figura 8: Dataframe Transposto dos dados coletados

	2019-04-23 t 00:00:07.500000- 03:00	2019-04-23 00:00:22.500000- 03:00	2019-04-23 00:00:37.500000- 03:00	2019-04-23 00:00:52.500000- 03:00	2019-04-23 00:01:07.500000- 03:00	2019-04-23 00:01:22.500000- 03:00	2019-04-23 00:01:37.500000- 03:00
Moinho1.Alimentacao.Correia.ligada	True	True	True	True	True	True	True
Moinho1.Alimentacao.TaxaMinerio.controlando	True	True	True	True	True	True	True
Moinho1.Alimentacao.TaxaMinerio.modos	True	True	True	True	True	True	True
Moinho1.Alimentacao.TaxaMinerio.mv	35.1681	35.1681	35.1681	35.131	34.9776	34.9073	34.8370
Moinho1.Alimentacao.TaxaMinerio.pv	109.291	109.45	111.269	111.867	111.684	110.585	109.48
Moinho1.Alimentacao.TaxaMinerio.pv.Duvidoso	False	False	False	False	False	False	False
Moinho1.Alimentacao.TaxaMinerio.pv.Ruim	False	False	False	False	False	False	False
Moinho1.Alimentacao.TaxaMinerio.spe	119.94	119.94	119.94	119.94	119.94	119.94	119.94

8 rows x 5760 columns

Fonte: do autor (2019)

Conforme a lista de variáveis necessárias apresentada anteriormente, temos:

- *Setpoint*: Moinho1.Alimentacao.TaxaMinerio.spe
- Variável manipulada: Moinho1.Alimentacao.TaxaMinerio.mv
- Variável controlada: Moinho1.Alimentacao.TaxaMinerio.pv
- Variável de status automático/manual: Moinho1.Alimentacao.TaxaMinerio.modos
- Variável de status de funcionamento da malha:  
Moinho1.Alimentacao.TaxaMinerio.controlando
- Variável de status de funcionamento do sensor:  
Moinho1.Alimentacao.TaxaMinerio.pv.Duvidoso e  
Moinho1.Alimentacao.TaxaMinerio.pv.Ruim

Num primeiro momento deve ser avaliada a consistência dos dados comparando o tipo de dado esperado e o tipo de dado recebido. As variáveis manipulada, controlada e os *setpoints* são variáveis numéricas, enquanto as outras (status de funcionamento da malha, status de funcionamento do sensor e status de automático/manual) são binárias (Verdadeiro ou Falso, 0 ou 1). Nota-se que os tipos de dados esperados estão condizentes com os recebidos.

O próximo passo é conferir a consistência dos valores recebidos. Não existe um procedimento padrão para essa avaliação, uma vez que cada variável representa algo completamente diferente da outra. Logo, adotou-se a seguinte metodologia para verificação preliminar dos valores:

Para variáveis numéricas, verificou-se se os valores estavam dentro de um range de valores esperados:

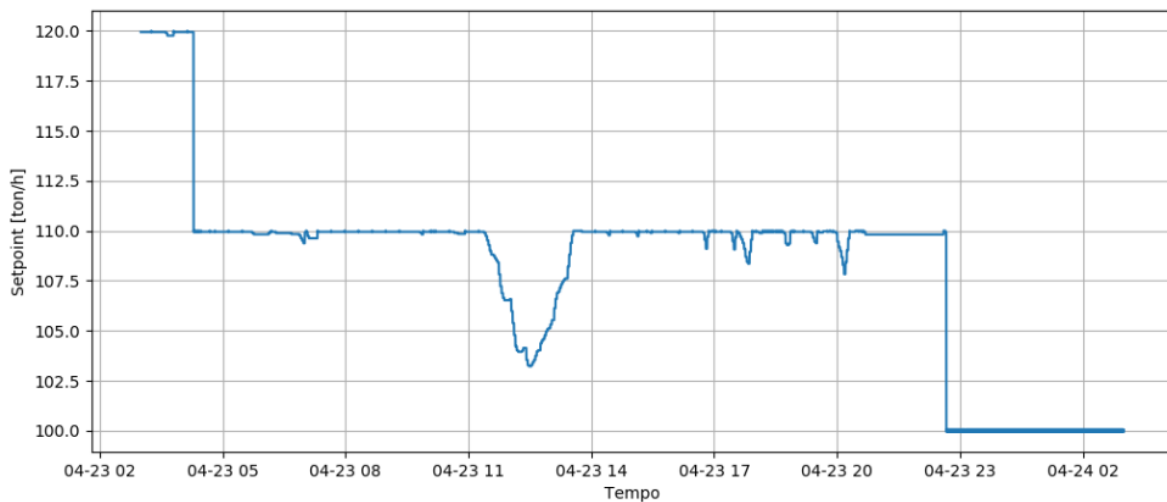
- 0 – 100 para a variável manipulada;
- 0 – 200 para a variável controlada e setpoint.

Para variáveis de estado, deve-se avaliar se a relação entre verdadeiro e falso está condizente com a operação.

#### 4.2.1 Setpoint, Setpoint máximo e Setpoint mínimo

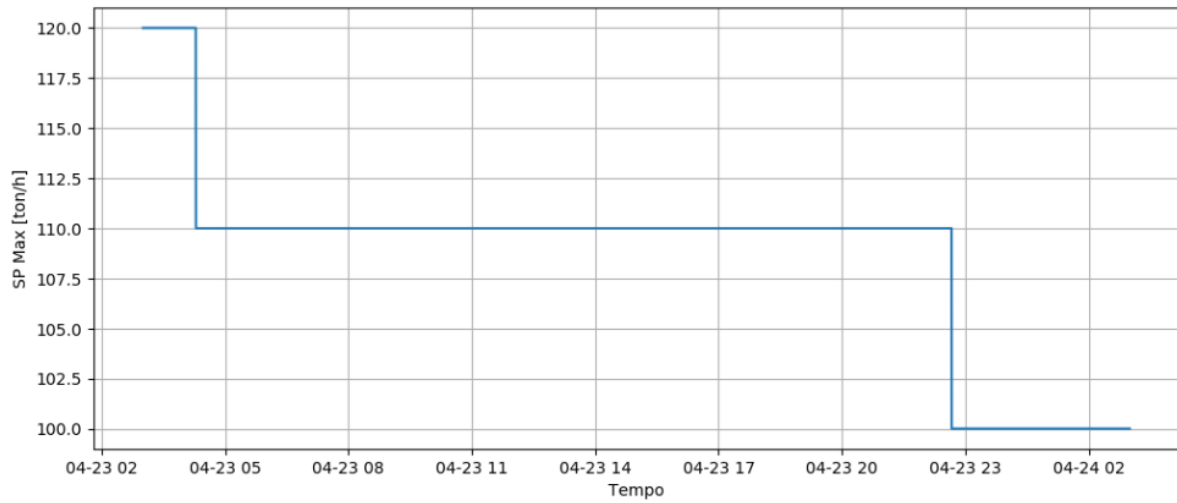
As figuras 9, 10 e 11 apresentam, respectivamente, a variação temporal do Setpoint, Setpoint máximo e Setpoint mínimo.

Figura 9: Gráfico Setpoint x Tempo



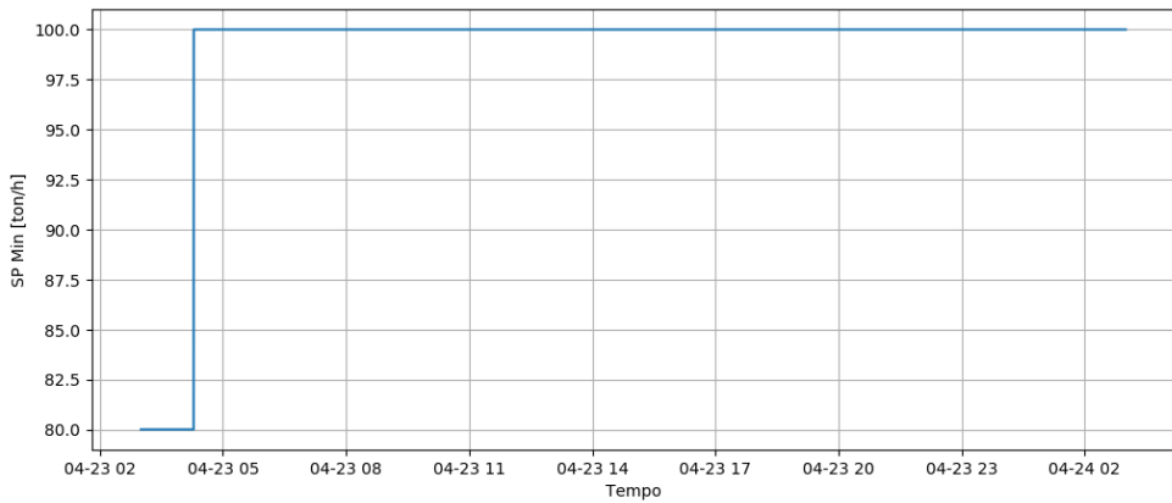
Fonte: do autor (2019)

Figura 10: Gráfico Setpoint Máximo x Tempo



Fonte: do autor (2019)

Figura 11: Gráfico Setpoint Mínimo x Tempo



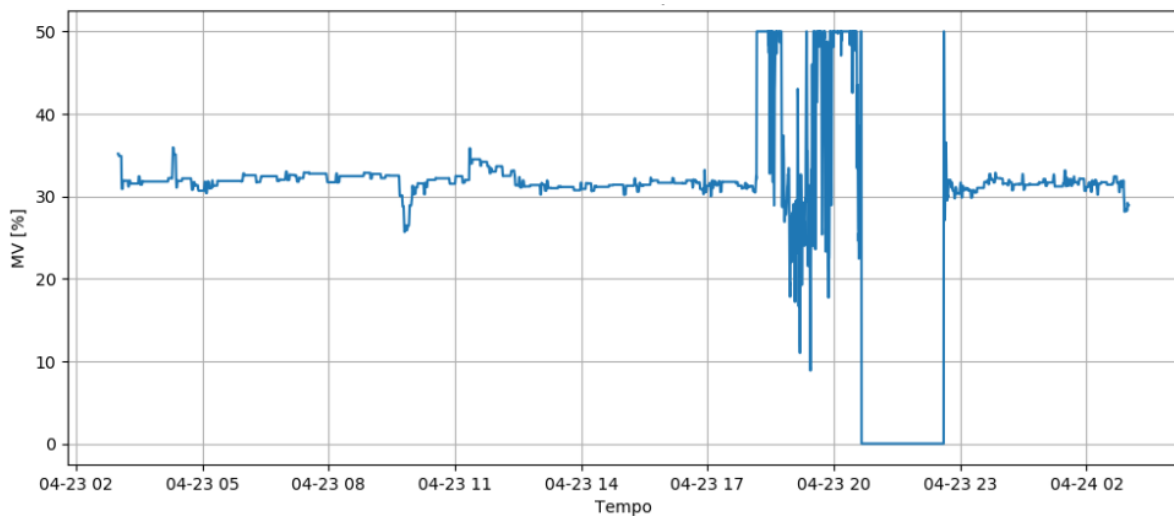
Fonte: do autor (2019)

Na figura 9, o setpoint apresenta variações, mas são provenientes do controle especialista, responsável por calcular o setpoint ótimo para o processo. Enquanto nas figuras 10 e 11, as variáveis apresentam um comportamento constante, uma vez que os valores são inseridos pelo operador. Por fim, os valores das três variáveis então dentro do range estipulado.

### 4.2.2 Variável Manipulada

De forma análoga, a variação temporal da variável manipulada está apresentada na figura 12, onde é possível observar um intervalo de tempo onde a variável apresenta uma oscilação abrupta seguida de uma queda do sinal para zero, possivelmente devido a um problema no processo que acarretou no desligamento do equipamento. Fora essa incidência, o valor do sinal da variável manipulada permaneceu dentro dos limites, variando entre 0 e 50.

Figura 12: Gráfico Variável Manipulada x Tempo

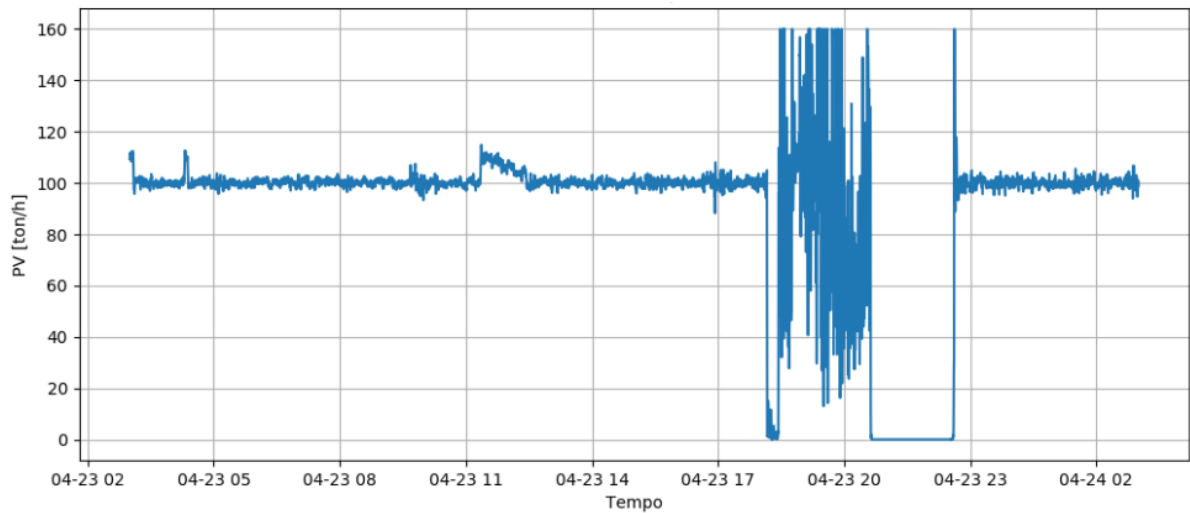


Fonte: do autor (2019)

### 4.2.3 Variável Controlada

Por fim, a última variável numérica também apresentou valores condizentes, variando entre 0 e 160, mas também apresentou a mesma oscilação abrupta seguida de uma queda do sinal para zero, reforçando a hipótese de um problema no processo, conforme a figura 13. O motivo para essa oscilação e conseqüente queda do sinal para zero pode ser devido a um problema no processo, que acarretou na parada da malha.

Figura 13: Gráfico Variável Controlada x Tempo

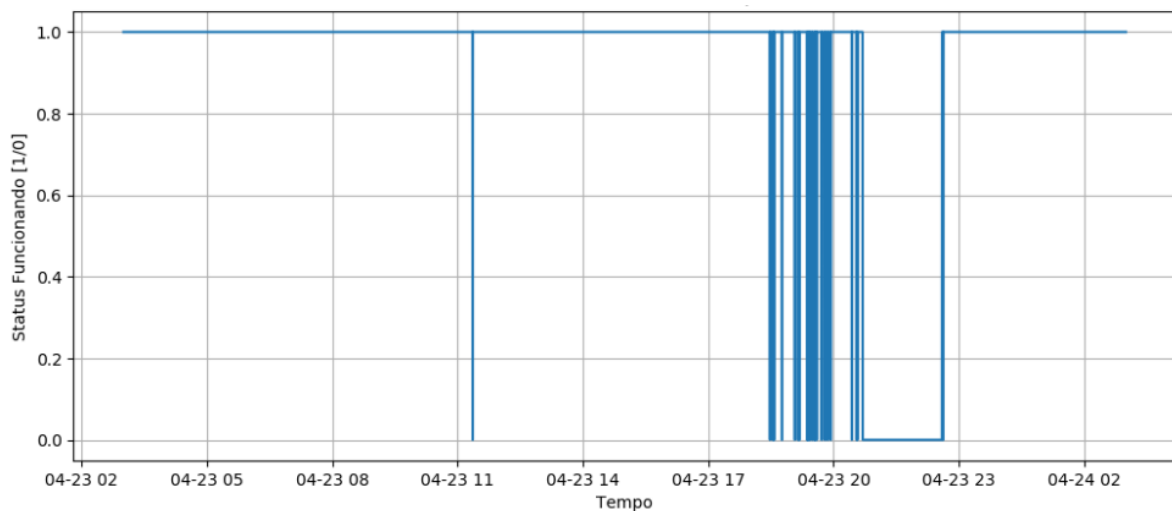


Fonte: do autor (2019)

#### 4.2.4 Variável de status de funcionamento da malha

Para a variável de status de funcionamento da malha apresentada na figura 14, a relação entre 1 (funcionando) e 0 (não funcionando), está condizente com o período de operação da malha e também com as medições da PV e MV observadas anteriormente (comportamento oscilatório seguido da queda do sinal para zero), acusando um desligamento da malha após o período oscilante.

Figura 14: Gráfico de Funcionamento da Malha x Tempo

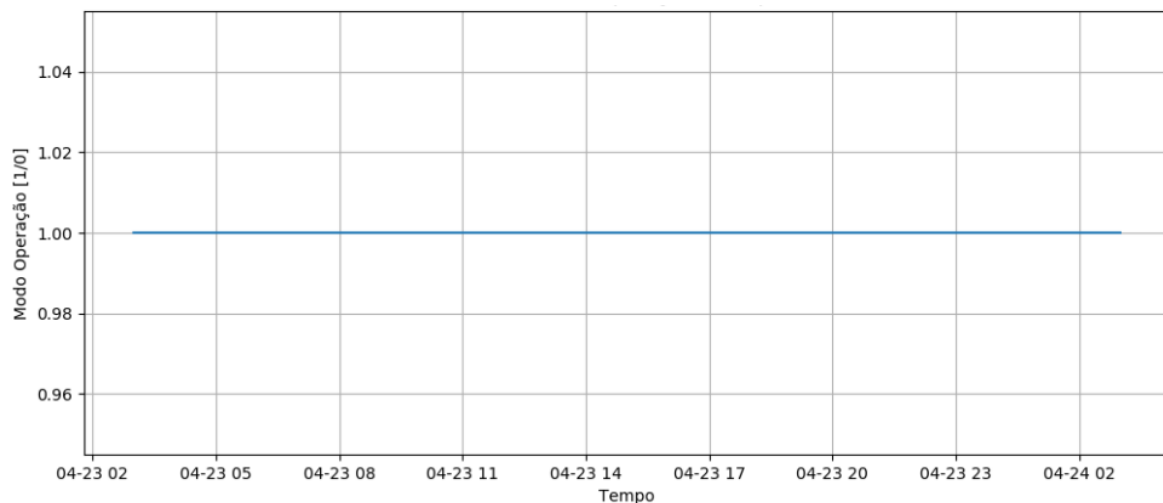


Fonte: do autor (2019)

#### 4.2.5 Variável de status automático/manual

A variável de status automático/manual (cujo comportamento está apresentado na figura 15) também está condizente com o modo de operação da malha e também apresenta uma característica que deve ser levada em consideração com relação ao tipo de análise a ser feita, onde a não operação da malha não influencia no modo de operação, o que pode gerar métricas incorretas (tempo em automático, por exemplo). Faz-se necessário o uso de filtros para esses dados, que serão abordados posteriormente.

Figura 15: Modo de Operação x Tempo

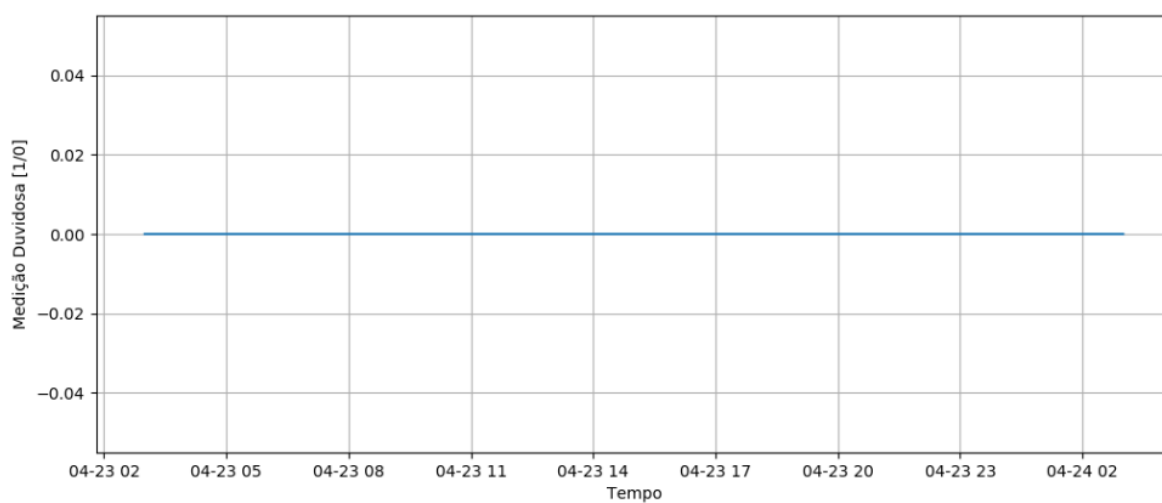


Fonte: do autor (2019)

#### 4.2.6 Variáveis de status de funcionamento do sensor

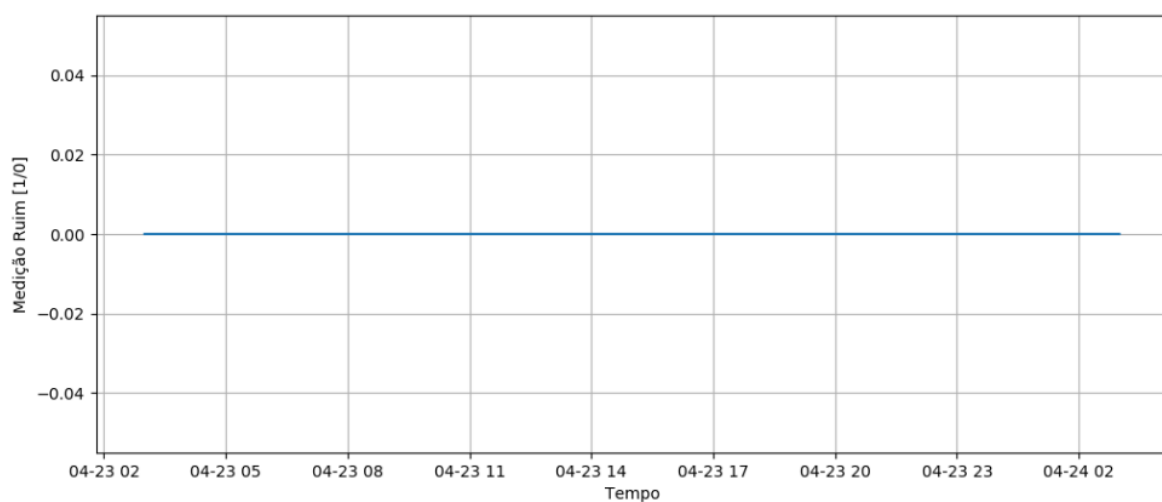
Existem duas variáveis de status de funcionamento do sensor, uma que indica uma medição ruim (indica que a PV não está recebendo valor ou está fora dos limites devido a um cabo desconectado, curto circuito, etc.) e o outro que indica uma medição duvidosa (indica que o valor da PV tem uma incerteza não conhecida, por exemplo offset errado, valor um pouco fora dos limites, etc.), o comportamento temporal das duas variáveis de status podem ser observadas nas figuras 16 e 17 respectivamente.

Figura 16: Medição Duvidosa x Tempo



Fonte: do autor (2019)

Figura 17: Medição Ruim x Tempo



Fonte: do autor (2019)

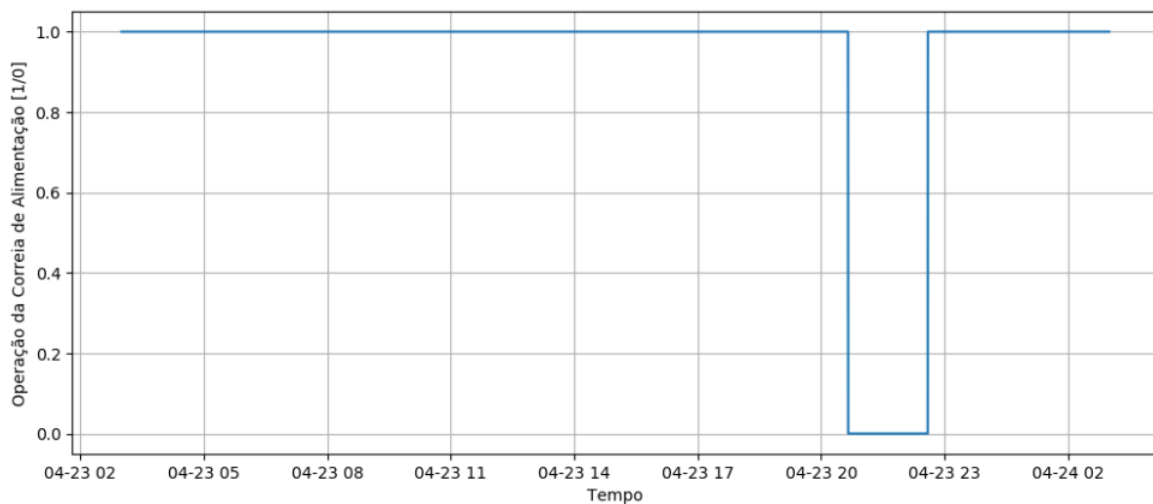
Apesar de nenhuma das duas variáveis acusar um comportamento estranho do sensor durante o intervalo onde a malha não estava operando, ainda que os valores medidos não estejam seguindo o mesmo comportamento apresentado anteriormente, esse comportamento pode ser advindo de o próprio fato da malha não estar operando. No entanto, deve ser usada uma outra estratégia para melhor avaliar o funcionamento do sensor, que também será abordada posteriormente.



## 4.2.7 Variável de status de funcionamento da correia de alimentação

Essa variável será utilizada como filtro para os dados, garantindo que as métricas serão calculadas num intervalo que representa uma operação normal da malha, ou seja, dados onde o valor dessa variável é igual a zero serão desconsiderados, conforme a figura 18.

Figura 18: Gráfico Funcionamento da correia de alimentação



Fonte: do autor (2019)

## 4.3 Implementação das métricas

### 4.3.1 Filtro de operação

Antes de calcular as métricas apresentadas, é importante definir quando o cálculo dessas métricas será condizente com a realidade, logo é necessário filtrar os dados por algum tipo de condição que garanta que o sistema esteja operando normalmente. O filtro utilizado pode depender de diversos fatores, como o funcionamento de outros equipamentos, valores mínimos de variáveis de processo, alguma variável de status genérica, etc. Para a malha desse trabalho, foi utilizado uma variável de status referente ao funcionamento da correia de alimentação e um valor mínimo para o medidor de vazão de minério igual a 40 [ton/h], e o código para obtenção desse filtro está apresentado na figura 19.

Figura 19: Filtro a ser aplicado nos dados

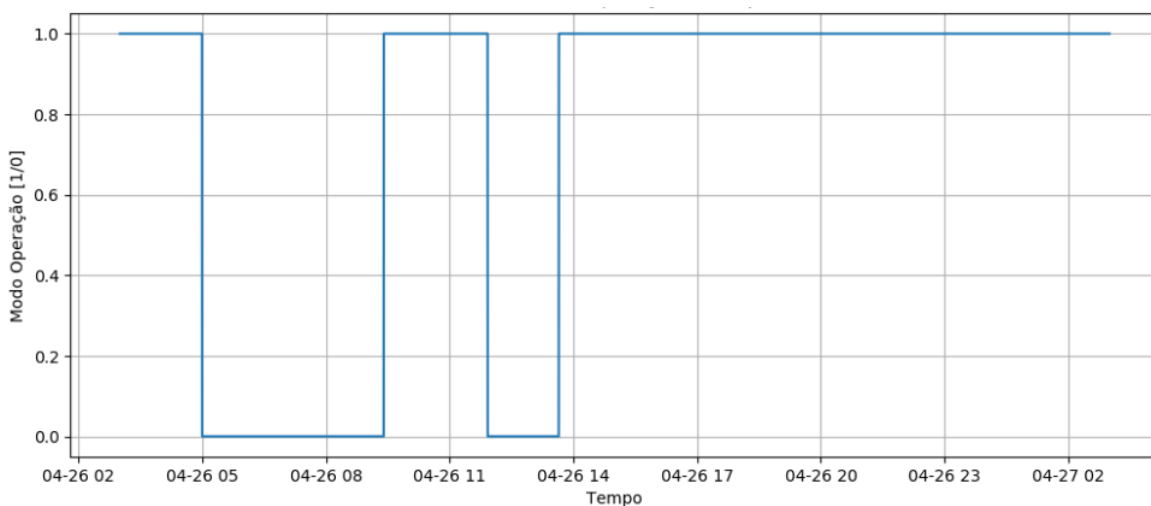
```
filtro_de_operacao = ((dados['Moinho1.Alimentacao.Correia.ligada'] > 0) &  
                    (dados['Moinho1.Alimentacao.TaxaMinerio.pv'] > 40))
```

Fonte: do autor (2019)

### 4.3.2 Mudanças de modo

Para obter o número de mudanças de modo, ou número de intervenções feitas pelo operador, foi necessário definir o que é considerada uma intervenção do operador. O modelo adotado foi que toda mudança na variável de status do modo de operação que fosse do valor verdadeiro para falso seria uma intervenção. Foi utilizado um intervalo de tempo diferente do apresentado na figura 15, uma vez que não apresentou nenhuma intervenção. A figura 20 apresenta dados referentes ao dia 26 de abril.

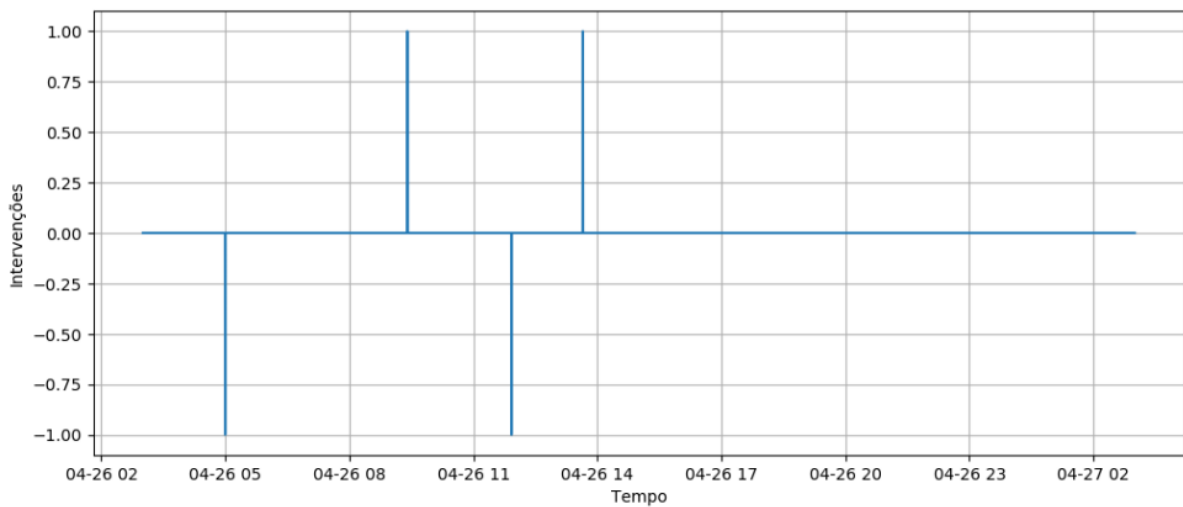
Figura 20: Gráfico do Modo de Operação x Tempo



Fonte: do autor (2019)

Observa-se duas intervenções. Utilizando a função `diff()`, que calcula a diferença cada valor da série de dados e o valor anterior, isso resulta em diversos valores iguais a zero, e nos momentos onde a variável de status vai do valor verdadeiro para falso e vice versa o valor será igual a 1 ou -1. Logo, é possível transformar essa série de dados para apresentar o modelo escolhido, representado na figura 21.

Figura 21: Gráfico de Intervenções x Tempo



Fonte: do autor (2019)

É possível então contabilizar de forma direta o número de intervenções contando o número de picos negativos. Essa contabilização é obtida filtrando a série resultante para valores menores que 0 e obtendo o número de incidências, cujo algoritmo é apresentado na figura 22.

Figura 22: Cálculo do número de intervenções

```
# Calculo do número de intervenções
intervencoes = ((dados['Moinho1.Alimentacao.TaxaMinerio.modos'][filtro_de_operacao]+0).diff() < 0).sum()
print('O número de intervenções é: '+str(intervencoes))

O número de intervenções é: 2
```

Fonte: do autor (2019)

### 4.3.3 Tempo em manual

Uma informação complementar à anterior é o tempo que o sistema operou em modo manual, ou seja, a duração da intervenção. Para esse cálculo existem diversas estratégias que podem ser adotadas. O método utilizado consistiu em extrair o índice do *dataframe* e transformá-lo numa série de intervalos de duração, ou seja, ao invés da série apresentar um timestamp, ela iria conter a diferença entre os *timestamp*. A figura 23 apresenta os cinco primeiros valores da série contendo a duração de cada intervalo.

Figura 23: Série de intervalos de duração (5 primeiros valores)

```
t
2019-04-26 00:00:07.500000-03:00    00:00:15.311059
2019-04-26 00:00:22.500000-03:00    00:00:15
2019-04-26 00:00:37.500000-03:00    00:00:15
2019-04-26 00:00:52.500000-03:00    00:00:15
2019-04-26 00:01:07.500000-03:00    00:00:15
Name: t, dtype: timedelta64[ns]
```

Fonte: do autor (2019)

Uma vez obtida a série de intervalos de duração, cria-se uma nova série booleana para filtrar a série anterior com base na condição desejada, que para a métrica a ser obtida consiste nos valores falsos. Por último, filtra-se a série de intervalos de duração com a série booleana e soma-se os valores, obtendo assim a duração (em horas) que a condição de interesse (malha em modo manual) foi atendida, segundo a figura 24.

Figura 24: Algoritmo para obtenção do tempo em manual

```
_tempo = dados.index.to_series().diff() #série com a duração dos intervalos
tempo = _tempo.fillna(_tempo.mean()) #inserção de possíveis dados faltantes
filtro = (dados['Moinho1.Alimentacao.TaxaMinerio.modelo'] < 1 & filtro_de_operacao) #filtro referente a condição de interesse
unidade_tempo = unidade_tempo=pd.Timedelta('1 h') #variavel para obter o valor da duração em horas
duracao = np.float64((np.nansum(tempo[filtro])) / unidade_tempo) #duração da condição de interesse
print('O tempo em manual é igual a: '+str(duracao)+' horas.')
```

O tempo em manual é igual a: 5.7625 horas.

Fonte: do autor (2019)

Observa-se que a malha operou em manual por aproximadamente 5 horas e 46 minutos.

#### 4.3.4 Saturação do atuador

O tempo de saturação do atuador pode ser calculado de forma análoga ao tempo em manual. O que diferirá será a condição do filtro a ser utilizado, que será o sinal da variável manipulada em 100% ou 0%. É possível também utilizar thresholds para esses valores de forma a limitar a operação. Foi utilizado um threshold de 10%, de forma que o tempo contabilizado foi referente a operação do atuador com sinal abaixo de 10% e o tempo acima de 90% (dois resultados distintos). Também foram utilizadas algumas métricas estatísticas para caracterizar a operação, como a média, o desvio padrão e percentis (5, 25, 50, 75 e 95), no entanto para obtê-las foi utilizado uma biblioteca chamada tdigest, instancia-se uma classe de forma que os objetos dessa classe armazenam informações que permitem o cálculo dessas métricas e também permite adicionar objetos dessa classe e obter as métricas referentes aos intervalos somados. A figura

25 apresenta o algoritmo para obtenção do tempo de saturação em horas, e a figura 26 apresenta os valores das métricas estatísticas referentes ao MV.

Figura 25: Algoritmo para obtenção do tempo de saturação do MV em horas

```

_tempo = dados.index.to_series().diff() #série com a duração dos intervalos
tempo = _tempo.fillna(_tempo.mean()) #inserção de possíveis dados faltantes
filtro_min = (dados['Moinho1.Alimentacao.TaxaMinerio.mv'] < 10) & filtro_de_operacao #filtro referente a condição de interesse
filtro_max = (dados['Moinho1.Alimentacao.TaxaMinerio.mv'] > 90) & filtro_de_operacao #filtro referente a condição de interesse
unidade_tempo = unidade_tempo=pd.Timedelta('1 h') #variavel para obter o valor da duração em horas
duracaoSatMin = np.float64((np.nansum(tempo[filtro_min])) / unidade_tempo) #duração da condição de interesse
duracaoSatMax = np.float64((np.nansum(tempo[filtro_max])) / unidade_tempo) #duração da condição de interesse
print('Tempo de operação saturado mínima: '+str(duracaoSatMin)+' horas')
print('Tempo de operação saturado máximo: '+str(duracaoSatMax)+' horas')

```

Tempo de operação saturado mínima: 0.0083333333333333 horas  
Tempo de operação saturado máximo: 0.0 horas

Fonte: do autor (2019)

Figura 26: Métricas estatísticas referentes ao MV em porcentagem

```

{'!stat(TaxaMinerio.mv)': <T-Digest: n=5186, centroids=416>,
 'avg(TaxaMinerio.mv)': 32.447013355156557,
 'std(TaxaMinerio.mv)': 4.2294400022790093,
 'cnt(TaxaMinerio.mv)': 5186,
 'p0(TaxaMinerio.mv)': 8.871545791625977,
 'p5(TaxaMinerio.mv)': 29.882098841667176,
 'p25(TaxaMinerio.mv)': 31.2819509150819,
 'p50(TaxaMinerio.mv)': 31.734947997128,
 'p75(TaxaMinerio.mv)': 32.43804201667692,
 'p95(TaxaMinerio.mv)': 40.88437211513513,
 'p100(TaxaMinerio.mv)': 50.0}

```

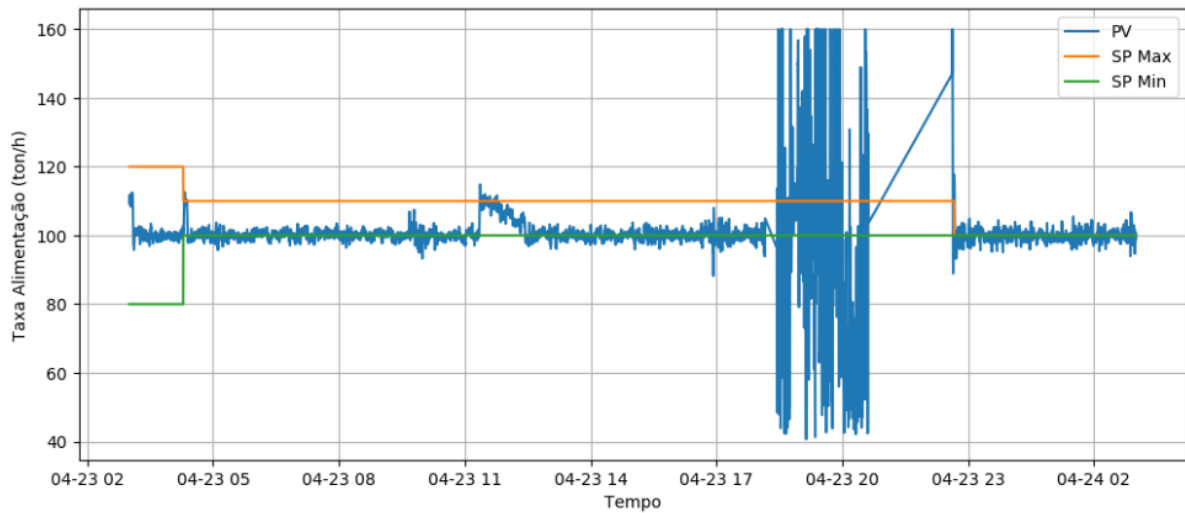
Fonte: do autor (2019)

A partir dos resultados, observa-se a malha não está operando nas extremidades, e esse conclusão pode ser reforçada com auxílio das métricas estatísticas, que apresentaram uma média de 32%, um desvio padrão de 4%.

### 4.3.5 Saturação do Controle

Uma métrica análoga à anterior e específica para sistemas que possuem limites de setpoint ou de operação, onde os limites irão variar em relação ao processo. Para o caso da malha analisada, o sistema especialista faz uso de limites de setpoint representadas por duas variáveis (limite máximo e limite mínimo). Na figura 27, estão apresentadas as variações temporais da PV, Setpoint máximo e setpoint mínimo.

Figura 27: Gráfico da variável controlada, set point máximo e mínimo



Fonte: do autor (2019)

É possível observar um grande intervalo de tempo em que a PV estava saturada no limite inferior, e um intervalo menor em que ela saturou no limite superior devido a uma prática discutível do operador ao igualar os limites. Além do tempo de saturação (apresentado na figura 28 juntamente com o algoritmo), também é interessante avaliar as mesmas métricas estatísticas para complementar essa análise, cujos valores estão presentes na figura 29.

Figura 28: Algoritmo para obtenção do tempo de saturação do PV em horas

```

_tempo = dados.index.to_series().diff() #série com a duração dos intervalos
tempo = _tempo.fillna(_tempo.mean()) #inserção de possíveis dados faltantes
filtro_max = ((dados['Moinho1.Alimentacao.TaxaMinerio.pv'] >= dados['Moinho1.Alimentacao.TaxaMinerio.spmaxOptprocess']) &
filtro_de_operacao)#filtro referente a condição de interesse
filtro_min = ((dados['Moinho1.Alimentacao.TaxaMinerio.pv'] <= dados['Moinho1.Alimentacao.TaxaMinerio.spmiOptprocess']) &
filtro_de_operacao)#filtro referente a condição de interesse
unidade_tempo = unidade_tempo=pd.Timedelta('1 h') #variavel para obter o valor da duração em horas
duracaoSatMin = np.float64((np.nansum(tempo[filtro_min])) / unidade_tempo) #duração da condição de interesse
duracaoSatMax = np.float64((np.nansum(tempo[filtro_max])) / unidade_tempo) #duração da condição de interesse
print('Tempo de operação saturado mínima: '+str(duracaoSatMin)+' horas')
print('Tempo de operação saturado máximo: '+str(duracaoSatMax)+' horas')

```

Tempo de operação saturado mínima: 9.81666583333 horas  
Tempo de operação saturado máximo: 2.8750055556 horas

Fonte: do autor (2019)

Figura 29: Métricas estatísticas da variável controlada

```
'!stat(TaxaMinerio.pv)': <T-Digest: n=5186, centroids=505>,  
'avg(TaxaMinerio.pv)': 99.884460442598566,  
'std(TaxaMinerio.pv)': 11.007095675720439,  
'cnt(TaxaMinerio.pv)': 5186,  
'p0(TaxaMinerio.pv)': 40.6866569519043,  
'p5(TaxaMinerio.pv)': 93.23178781191508,  
'p25(TaxaMinerio.pv)': 99.02277374267578,  
'p50(TaxaMinerio.pv)': 100.04846236573836,  
'p75(TaxaMinerio.pv)': 101.21055013191798,  
'p95(TaxaMinerio.pv)': 109.23956460952759,  
'p100(TaxaMinerio.pv)': 160.0}
```

Fonte: do autor (2019)

#### 4.3.6 Erro médio

O erro médio será calculado somando os módulos das diferenças absolutas dividido pela contagem de amostras. São utilizadas duas variáveis para que o cálculo possa ser estendido para intervalos maiores, seguindo a estratégia mencionada anteriormente para seccionamento dos dados. Logo, caso o intervalo seja referente a um dia, será a soma das somas horárias dividida pela soma das contagens, uma vez que a média das médias não seria equivalente.

Figura 30: Cálculo do erro médio

```
erro = dados['Moinho1.Alimentacao.TaxaMinerio.pv'] - dados['Moinho1.Alimentacao.TaxaMinerio.spe'] # Serie de erro  
soma_erro = erro[filtro_de_operacao].abs().sum() # Soma dos erros  
contagem = erro[filtro_de_operacao].count() # Contagem das amostras  
erro_medio = soma_erro/contagem # Erro médio  
print('Erro médio: '+str(erro_medio)+' ton/h')
```

Erro médio: 10.1179259323 ton/h

Fonte: do autor (2019)

A figura 30 mostra o algoritmo para o cálculo do erro médio, e o resultado obtido para o intervalo é igual a 10 [ton/h]. Esse valor é subjetivo ao processo, e deve ser analisado de forma a determinar se essa variação entre a PV e o setpoint é aceitável ou não.

#### 4.3.7 Tempo de operação da área

Contabilizando o tempo em que o filtro de operação é verdadeiro, é possível obter o tempo de operação da área onde a malha se encontra. Essa métrica não apresenta uma informação sobre a malha de controle propriamente dita, mas mostra um indicativo sobre o processo como um todo. O valor desse tempo será calculado conforme descrito na figura 31.

Figura 31: Algoritmo para obtenção do tempo de operação da área

```
_tempo = dados.index.to_series().diff() #série com a duração dos intervalos
tempo = _tempo.fillna(_tempo.mean()) #inserção de possíveis dados faltantes
filtro = filtro_de_operacao #filtro referente a condição de interesse
unidade_tempo = unidade_tempo=pd.Timedelta('1 h') #variavel para obter o valor da duração em horas
duracao = np.float64((np.nansum(tempo[filtro])) / unidade_tempo) #duração da condição de interesse
print('O tempo de operação da área é igual a: '+str(duracao)+' horas.')
```

O tempo de operação da área é igual a: 21.6083338889 horas.

Fonte: do autor (2019)

#### 4.3.8 Tempo de operação do sensor

O tempo de operação do sensor é representado pelo tempo em que as variáveis de status de funcionamento do sensor: PV ruim e PV duvidoso, apresentam valor Falso. O algoritmo será o mesmo utilizado para obtenção das outras métricas de tempo, e o filtro de dados será o filtro de operação da área e a condição de funcionamento do sensor (PV ruim = Falso e PV duvidoso = Falso), conforme a figura 32.

Figura 32: Algoritmo para obtenção do tempo de operação do sensor

```
_tempo = dados.index.to_series().diff() #série com a duração dos intervalos
tempo = _tempo.fillna(_tempo.mean()) #inserção de possíveis dados faltantes
filtro = ((dados['Moinho1.Alimentacao.TaxaMinerio.pv.Duvidoso'] == False) &
          (dados['Moinho1.Alimentacao.TaxaMinerio.pv.Ruim'] == False) &
          filtro_de_operacao) #filtro referente a condição de interesse
unidade_tempo = unidade_tempo=pd.Timedelta('1 h') #variavel para obter o valor da duração em horas
duracao = np.float64((np.nansum(tempo[filtro])) / unidade_tempo) #duração da condição de interesse
print('O tempo de operação do sensor é igual a: '+str(duracao)+' horas.')
```

O tempo de operação do sensor é igual a: 21.6083338889 horas.

Fonte: do autor (2019)

#### 4.3.9 Normalização das métricas de tempo

Como último passo, foi utilizada uma normalização das métricas temporais, ou seja, dividiu-se a quantidade de horas obtida pela quantidade total de horas referente ao intervalo analisado. Logo, quando for analisado um intervalo de um dia, ao invés de ser apresentado um valor referente a uma quantidade de horas, será apresentado um valor percentual que representará a porcentagem de horas referente ao intervalo, com isso é possível abstrair a relação entre uma quantidade de horas e a quantidade de horas presente no intervalo.

A quantidade de horas do intervalo não será a quantidade total, mas sim a quantidade de horas em operação (representada pelo tempo de operação da área).

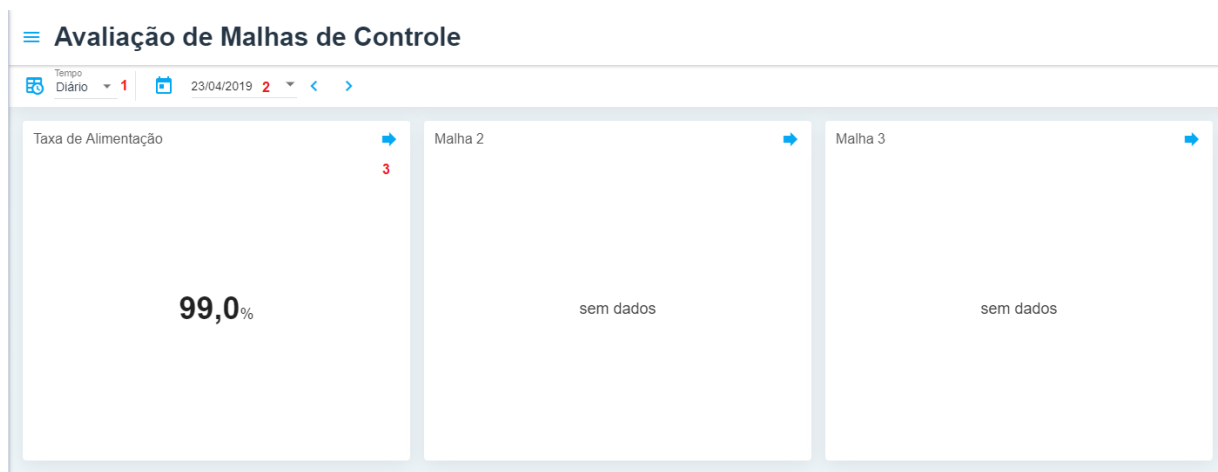


#### 4.4 Apresentação dos resultados

Foi utilizado o framework para frontend da empresa Optimus. A estrutura consiste numa página web onde são apresentados diversos conjuntos de *dashboards* ou *dashboard sets*. Os *dashboards* por sua vez são estruturas matriciais (divididas em linhas e colunas) que serão preenchidas com as informações a serem apresentadas, sejam elas gráficos, valores, botões, etc., que serão geradas utilizando funções já existentes do framework. Os dados a serem apresentados serão referentes às métricas calculadas para diferentes períodos e intervalos de tempo.

A estratégia de apresentação consiste num menu principal onde estarão dispostas as malhas analisadas conforme a figura 33.

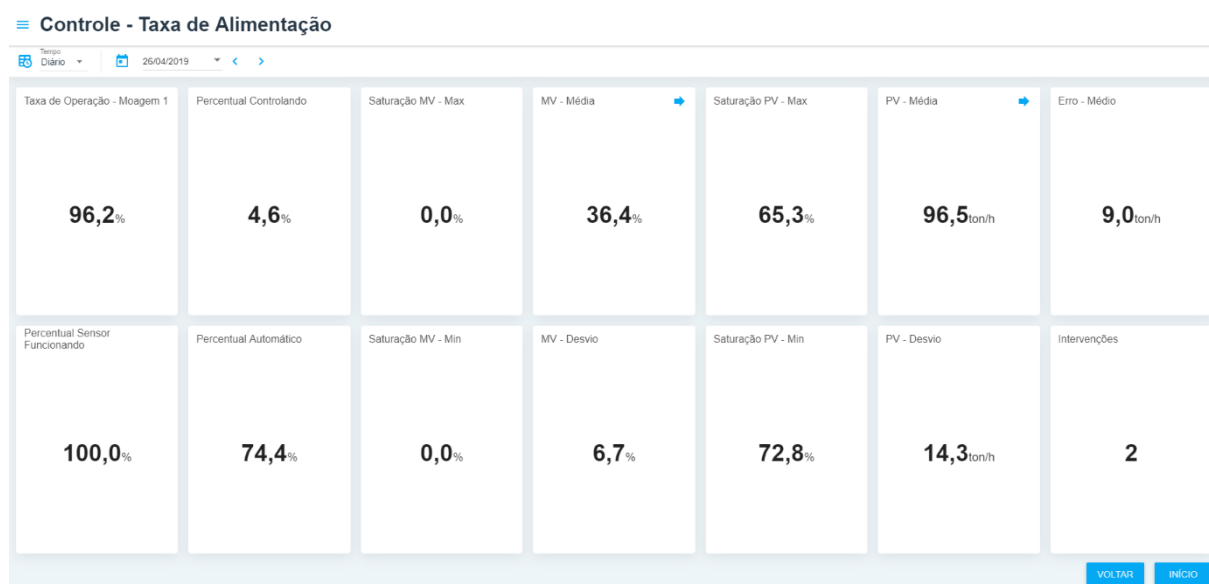
Figura 33: Tela de apresentação das malhas



Fonte: do autor (2019)

O campo com número 1 permite alterar o período analisado (diário, mensal ou anual), 2 é o campo para alteração da data de análise, e 3 o bloco onde está apresentado a identificação da malha, o percentual de operação em automático referente ao período, e ao clicar na seta azul, o usuário será redirecionado à tela onde serão apresentadas as outras métricas referentes a essa malha, mostrada na figura 34.

Figura 34: Tela de métricas da malha de controle



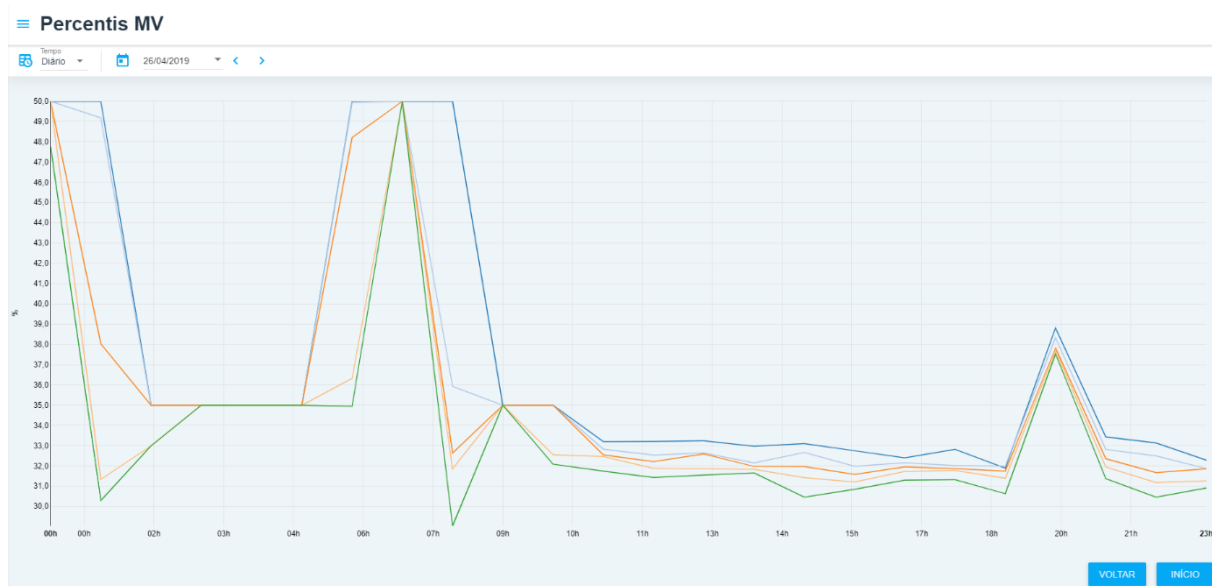
Fonte: do autor (2019)

A tela também dispõe de campos para seleção do tipo de intervalo e a data do intervalo, e apresenta quadros com os valores calculados de cada métrica e respectivas unidades:

- Taxa de Operação – Moagem 1 [%]: taxa de tempo de operação da área;
- Percentual Sensor Funcionando [%]: taxa de tempo de funcionamento do sensor;
- Percentual Controlando [%]: taxa de tempo do controle especialista operando;
- Percentual Automático [%]: taxa de tempo da malha em automático;
- Saturação MV – Max [%]: taxa de tempo de saturação da MV no limite superior;
- Saturação MV – Min [%]: taxa de tempo de saturação da MV no limite inferior;
- MV – Média [%]: valor médio da MV referente ao intervalo;
- MV – Desvio [%]: valor do desvio padrão da MV referente ao intervalo;
- Saturação PV – Max [ton/h]: taxa de tempo de saturação da PV no Setpoint superior;
- Saturação PV – Min [ton/h]: taxa de tempo de saturação da PV no Setpoint inferior;
- PV – Média [ton/h]: valor médio da PV referente ao intervalo;
- PV – Desvio [ton/h]: valor do desvio padrão da PV referente ao intervalo;
- Erro – Médio [ton/h]: valor médio da diferença entre a PV e o Setpoint;
- Intervenções [un]: número de intervenções do operador.

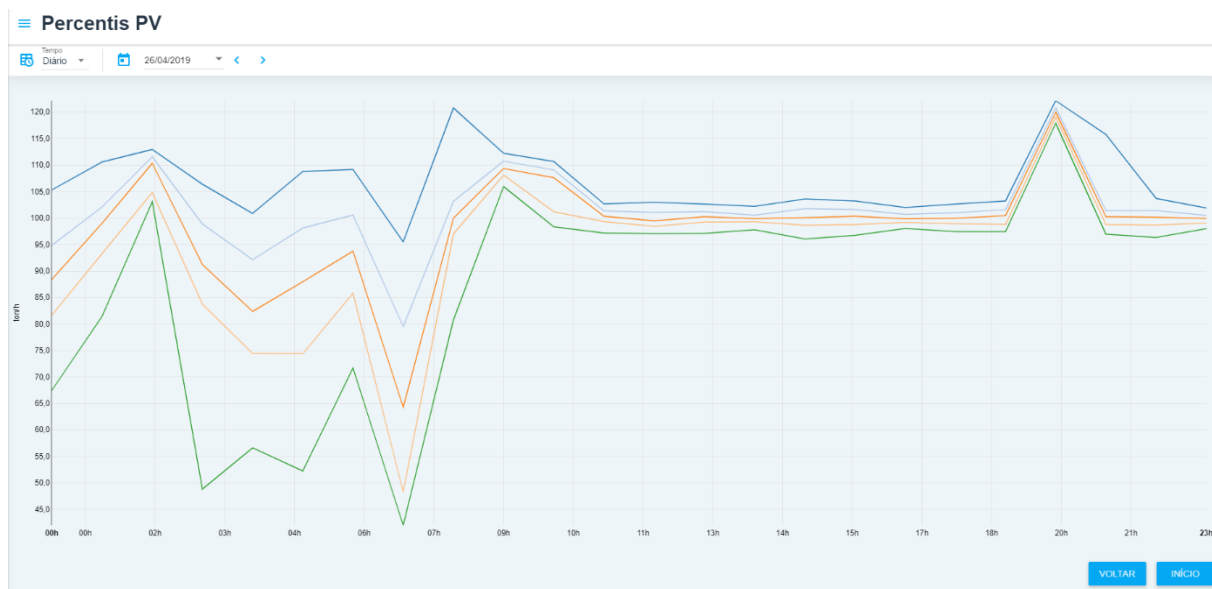
Existem também duas setas azuis, no quadro referente à média do MV e à média da PV, que redirecionam aos gráficos dos percentis das respectivas variáveis. Os gráficos da MV e da PV estão apresentados nas figuras 35 e 36, respectivamente.

Figura 35: Gráfico de percentis da variável manipulada



Fonte: do autor (2019)

Figura 36: Gráfico de percentis da variável controlada



Fonte: do autor (2019)

Os gráficos apresentam os valores dos percentis 5, 25, 50, 75 e 95 referentes a cada intervalo horário de dados de cada variável. O valor do desvio padrão apresenta a dispersão dos dados, enquanto os valores de percentis apresentados no gráfico podem ser utilizados para observar a real operação da malha, no sentido de averiguar a faixa de operação tanto em relação aos valores da MV quanto da PV. Com isso, é possível concluir se a distribuição dos valores está adequada à operação ou não.

## 5 CONCLUSÃO

O estudo proposto visou investigar métricas para avaliar o desempenho de malhas de controle PID, visto que existem inúmeras malhas de controle em plantas industriais e são parte fundamental dos processos. Caso elas não estejam operando corretamente, pode acarretar em um produto final com qualidade inferior, desperdício de matéria prima e desgaste prematuro de componentes da malha, como atuadores, sensores, etc. E uma vez obtida uma avaliação sobre o desempenho da malha, possibilita o profissional capacitado a realizar eventuais correções ou mudanças na malha a fim de melhorar a qualidade do processo e evitar os possíveis problemas citados anteriormente. Além disso, também permitiu a incorporação de conceitos provenientes da Indústria 4.0, que estão cada vez mais presentes no contexto industrial atual, como a computação em nuvem e descentralização da informação, e que farão parte fundamental da operação das indústrias existentes que pretendem otimizar suas operações.

A pesquisa bibliográfica de índices de desempenho apresentou uma certa tendência para métricas que avaliam aspectos relacionados à presença e classificação de oscilações em malhas de controle requerendo um conhecimento sobre modelagem e caracterização do sistema, enquanto as ferramentas presentes no mercado tendiam para a análise da operação da malha, visando avaliar aspectos que englobam como a malha está sendo utilizada. Foi dado uma maior ênfase às últimas, uma vez que o conteúdo desse estudo se tornaria muito abrangente caso todas fossem implementadas. No entanto, todas as métricas fornecem informações interessantes e distintas sobre a malha de controle, e conseqüentemente permitem ações corretivas distintas, cabendo ao profissional capacitado avaliar qual métrica deve ser priorizada.

As métricas calculadas forneceram informações pertinentes ao desempenho da malha de controle e a forma de exibição se mostrou visualmente intuitiva e de fácil utilização, e as mesmas podem ser conferidas para diferentes intervalos e datas, sendo possível comparar valores históricos e observar a evolução das métricas com o tempo e caso alguma alteração na malha seja feita, comparar o resultado dessa possível alteração com o estado anterior.

É possível também avaliar outras malhas caso elas possuam variáveis equivalentes às necessárias para os cálculos das métricas, e também implementar novas métricas, desde que os cálculos sejam factíveis. A metodologia apresentada poderá ser seguida independente da malha, uma vez que as métricas não dependem do tipo de malha de controle, desde que ela possua as variáveis necessárias para o cálculo. No entanto, a forma de validação de algumas das variáveis

ainda está sujeita ao conhecimento do processo, apesar de ser uma informação relativamente simples de ser obtida.

Essa estratégia de avaliação é, inicialmente, apenas para prover um diagnóstico rápido da operação da malha, possibilitando o profissional capacitado fazer possíveis intervenções com base nesses resultados. É possível também, uma vez que sejam estabelecidos parâmetros de operações para essas métricas, adicionar valores de *benchmarking* para expor quão bem a malha está operando em relação aos valores esperados. Com isso, seria possível implementar uma lógica no código de apresentação dos resultados para mostrar os números em vermelho ou verde, por exemplo, caso as métricas estejam abaixo ou acima, respectivamente, dos valores de *benchmarking*.

## 6 BIBLIOGRAFIA

DE CAMPOS, Mario Cesar M. Massa; TEIXEIRA, Herbert CG. **Controles típicos de equipamentos e processos industriais**. Edgard Blücher, 2010.

HÄGGLUND, Tore. A control-loop performance monitor. **Control Engineering Practice**, v. 3, n. 11, p. 1543-1551, 1995

HÄGGLUND, Tore. Automatic detection of sluggish control loops. **Control Engineering Practice**, v. 7, n. 12, p. 1505-1511, 1999.

HARRIS, Thomas J. Assessment of control loop performance. **The Canadian Journal of Chemical Engineering**, v. 67, n. 5, p. 856-861, 1989.

HORCH, Alexander. A simple method for detection of stiction in control valves. **Control Engineering Practice**, v. 7, n. 10, p. 1221-1231, 1999.

JELALI, Mohieddine. **Control performance management in industrial automation: assessment, diagnosis and improvement of control loop performance**. Springer Science & Business Media, 2012.