Dimensionamento Control-Aware de Controladores Compartilhados em Ambientes de Borda Kubernetes para Sistemas IIoT

Marcelo A. C. Fernandes[1]

[1]InovAI Lab – nPITI/IMD – UFRN
Leading Advanced Technologies Center of Excellence (LANCE) – nPITI/IMD – UFRN
Department of Computer Engineering and Automation (DCA) – UFRN
Natal – RN – Brazil

mfernandes@dca.ufrn.br

Abstract. *Este trabalho investiga o dimensionamento de controladores fuzzy compartilhados em ambientes de computação de borda baseados em Kubernetes para sistemas IIoT com controle em tempo real. A partir de uma análise experimental envolvendo diferentes plantas, regimes operacionais e configurações de carga, é caracterizado o compromisso entre qualidade de controle (QoC), requisitos temporais e custo computacional. Os resultados mostram que métricas tradicionais de computação são insuficientes para capturar limites críticos do sistema, sendo métricas temporais baseadas em percentis fundamentais para identificar regiões de operação inviáveis. Com base nesses achados, é proposta uma política de dimensionamento control-aware capaz de selecionar o número mínimo de controladores que satisfaz simultaneamente restrições de QoC e tempo real, com maior eficiência em relação a estratégias convencionais.*

1. Introduction

A crescente adoção de sistemas de Internet Industrial das Coisas (IIoT) tem impulsionado a execução de múltiplas malhas de controle em tempo real sobre infraestruturas computacionais distribuídas, particularmente em ambientes de computação de borda. Nesse contexto, dispositivos físicos heterogêneos passam a depender de serviços de controle executados remotamente, sujeitos a limitações de recursos computacionais, variabilidade de carga, concorrência e atrasos de comunicação. Diferentemente de arquiteturas tradicionais de controle dedicadas, nas quais cada planta dispõe de um controlador exclusivo, ambientes de borda modernos frequentemente adotam modelos de controle compartilhado, nos quais um conjunto reduzido de controladores atende simultaneamente múltiplos dispositivos, introduzindo novos desafios para a garantia de qualidade de controle e requisitos de tempo real [Kuchuk and Malokhvii 2024, Yan et al. 2024, Sadikhov et al. 2025, Caiazza et al. 2025].

A incorporação de plataformas de orquestração baseadas em contêineres, como o Kubernetes (K8s), amplia ainda mais a complexidade desse cenário ao introduzir elasticidade dinâmica, escalonamento automático e compartilhamento explícito de recursos computacionais. Embora tais mecanismos sejam eficazes para aplicações orientadas a throughput ou latência média, seu uso em sistemas de controle em tempo real é não trivial, uma vez que decisões de escalonamento podem impactar diretamente o fechamento da malha de controle. Em particular, a variação dinâmica do número de controladores ativos,

a concorrência entre requisições de controle e a interferência entre cargas computacionais podem induzir atrasos, jitter e violações temporais que comprometem a estabilidade e o desempenho do sistema controlado [Li et al. 2021, Silva et al. 2024, Kaur et al. 2020].

Apesar do avanço significativo em estratégias de escalonamento horizontal para aplicações em nuvem e borda, a maioria das abordagens existentes baseia-se predominantemente em métricas tradicionais de computação, como uso médio de CPU, memória ou taxa de requisições [Hall et al. 2024]. No contexto de sistemas IIoT com controle em tempo real, tais métricas são insuficientes para capturar restrições críticas impostas pela aplicação de controle, como limites máximos de atraso, variações temporais extremas e degradação da qualidade de controle. Como consequência, políticas de escalonamento convencionais podem manter o sistema computacionalmente estável enquanto violam requisitos temporais fundamentais para o correto funcionamento das malhas de controle.

Diante desse cenário, torna-se necessário repensar o dimensionamento de controladores em ambientes de borda a partir de uma perspectiva sensível à aplicação de controle, na qual decisões de escalonamento considerem explicitamente métricas de qualidade de controle (QoC), além de requisitos de tempo real. Este trabalho investiga esse problema no contexto de controladores fuzzy compartilhados executando em K8s, analisando sistematicamente o impacto do compartilhamento de controladores sobre a QoC, o comportamento temporal e o custo computacional. A partir dessa análise, o artigo estabelece as bases para uma política de dimensionamento *control-aware*, capaz de selecionar configurações eficientes de controle compartilhado sem comprometer os requisitos fundamentais do sistema.

2. Configuração Experimental e Geração do Dataset

Os experimentos foram conduzidos em um ambiente de borda baseado em K8s, no qual os controladores fuzzy são executados como *pods* escalonáveis, enquanto os dispositivos IIoT e as plantas físicas são emulados por aplicações independentes. A arquitetura experimental adotada é ilustrada na Figura 1, que apresenta a organização lógica e funcional do sistema de controle compartilhado em borda. Conforme mostrado na Figura 1, o sistema é composto por um conjunto de máquinas de inferência fuzzy, denominadas *Fuzzy Inference Machines* (FIMs), executando em um cluster K8s no servidor MEC. Cada FIM implementa um controlador fuzzy do tipo Fuzzy-PI e é capaz de atender, de forma concorrente, múltiplos dispositivos IIoT, caracterizando um cenário de controle compartilhado. O número de FIMs ativas no sistema em um dado instante é denotado por K_n, refletindo o fato de que a quantidade de controladores disponíveis pode variar dinamicamente ao longo do tempo em função de decisões de escalonamento no ambiente K8s. Ao longo dos experimentos, K_n assume valores discretos pertencentes ao conjunto K, definido a seguir.

Cada dispositivo IIoT está associado a uma planta física emulada e executa localmente o fechamento da malha de controle, comunicando-se remotamente com uma das FIMs disponíveis. A interação ocorre por meio de requisições HTTP, nas quais o dispositivo envia periodicamente o erro de controle $e(n)$ e sua derivada $\dot{e}(n)$, recebendo como resposta o sinal de controle $u(n)$ calculado pela FIM. Esse mecanismo explicita a separação entre a dinâmica da planta, a lógica de controle e a infraestrutura computacional, permitindo isolar de forma controlada os efeitos de computação, rede e concorrência sobre o

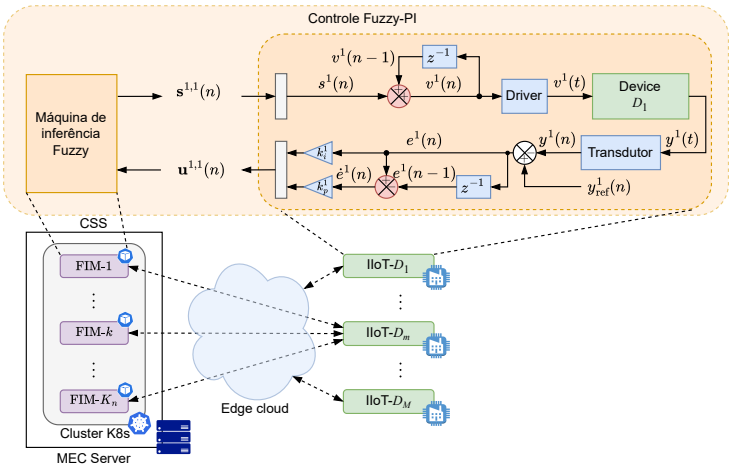


Figure 1. Arquitetura experimental do sistema de controle fuzzy compartilhado em ambiente de borda.

desempenho do sistema de controle.

Foram consideradas três plantas com dinâmicas distintas, representativas de diferentes classes de sistemas de controle [Silva et al. 2019]. O motor DC representa um sistema com dinâmica rápida e alta sensibilidade a atrasos, sendo avaliado nos setpoints de 500, 900, 1200 e 1500 rpm. O tanque de controle de nível representa um sistema de dinâmica lenta e naturalmente amortecida, avaliado nos setpoints de 10, 15, 20 e 25 unidades de nível. A planta veicular representa um sistema de controle de velocidade com elevada inércia física, avaliado nos setpoints de 50, 80, 120 e 150 km/h. A escolha dessas plantas permite cobrir um amplo espectro de comportamentos dinâmicos, viabilizando a análise do impacto do compartilhamento de controladores em diferentes regimes de operação.

Para cada combinação de planta e setpoint, o número de dispositivos IIoT ativos foi variado em $M \in \{1, 2, 4, 8, 12, 16, 20, 24, 28, 32\}$, enquanto o número de controladores compartilhados disponíveis no sistema foi configurado em $K \in \{1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 16\}$. Cada par (M, K) define um cenário distinto de concorrência e compartilhamento de recursos computacionais, sendo que, em cada experimento, assume-se $K_n = K$ constante ao longo da execução. No total, foram executados 1440 experimentos independentes, correspondentes a todas as combinações de planta, setpoint, M e K.

Durante cada experimento, foram coletadas métricas em diferentes níveis do sistema. As métricas de controle incluem o erro absoluto integrado (*Integral Absolute Error* – IAE), o tempo de loop de controle e o atraso de requisição, calculados a partir dos sinais registrados pelos dispositivos IIoT. As métricas computacionais correspondem ao uso de CPU das FIMs, monitorado em nível de *pod* no K8s. As métricas de rede incluem o número de pacotes e o volume de tráfego de entrada e saída associados às requisições de controle. As métricas temporais foram agregadas utilizando percentis, em particular o percentil 90 (P_{90}), de modo a capturar comportamentos extremos relevantes para sistemas de controle em tempo real, enquanto as métricas de qualidade de controle foram

agregadas por média ao longo dos dispositivos IIoT. A partir dos dados brutos, foi construído um dataset formal consolidado, no qual cada linha representa um experimento completo. Esse processo envolveu a agregação das métricas de controle, computação e rede, o cálculo de estatísticas descritivas (média, mediana e percentis), a associação explícita entre métricas e parâmetros experimentais $(\text{planta}, \text{setpoint}, M, K)$, bem como a validação de consistência e a remoção de entradas inválidas. O dataset gerado está publicamente disponibilizado em [Fernandes 2026].

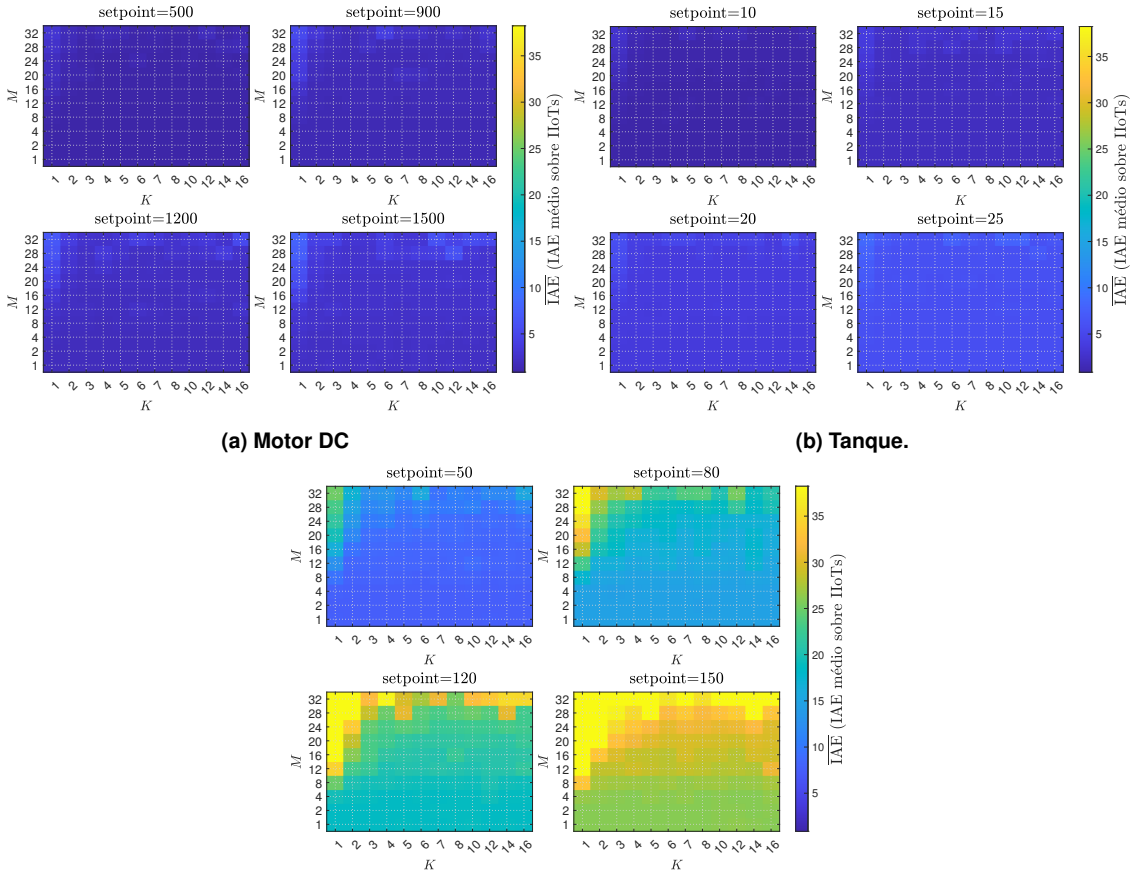
3. Análise Experimental do Espaço de Operação

O objetivo desta análise é identificar padrões estruturais no espaço (M, K), compreender como diferentes classes de dinâmicas físicas respondem ao compartilhamento de controladores e caracterizar os limites operacionais impostos por requisitos de QoC, tempo real e custo computacional. Em particular, a análise busca evidenciar as regiões de operação viáveis e inviáveis associadas a diferentes combinações de carga e compartilhamento, revelando transições abruptas induzidas por violações temporais que não são capturadas por métricas médias de desempenho, mesmo quando a QoC permanece dentro de níveis aceitáveis. Os resultados obtidos nesta seção fornecem a base quantitativa para a definição de um valor mínimo de controladores compartilhados, denotado por $K_{\min}$, capaz de satisfazer simultaneamente restrições de QoC e de tempo real ao longo de todo o conjunto de cargas consideradas. Essa caracterização do espaço (M, K) fundamenta diretamente a formulação da política de dimensionamento *control-aware* apresentada na Seção 4, bem como o algoritmo proposto para a determinação sistemática de $K_{\min}$.

3.1. Qualidade de Controle (IAE)

As Figuras 2a, 2b e 2c apresentam os mapas de calor do IAE médio sobre todos os IIoTs, $\overline{\text{IAE}}$, em função de (M, K) para as plantas motor DC, tanque e veículo, respectivamente, considerando os quatro setpoints de cada planta.

Para o motor DC (ver Figura 2a), o $\overline{\text{IAE}}$ apresenta baixa sensibilidade ao número de controladores para valores moderados de K, permanecendo relativamente estável para $K \geq 4$ em todos os setpoints. O aumento do setpoint resulta em um deslocamento gradual do $\overline{\text{IAE}}$ para valores mais altos, especialmente para grandes valores de M, indicando maior esforço de controle. No entanto, a degradação ocorre de forma suave, sem regiões abruptas de instabilidade. No caso do tanque (ver Figura 2b), o $\overline{\text{IAE}}$ é ainda menos sensível ao compartilhamento. Em todos os setpoints analisados (10, 15, 20 e 25), o $\overline{\text{IAE}}$ permanece baixo e praticamente constante em todo o espaço (M, K). Esse comportamento reflete a dinâmica lenta e naturalmente amortecida do sistema de nível, que tolera variações de latência e concorrência sem impacto significativo na qualidade média de controle. A planta associada ao veículo (ver Figura 2c) apresenta um comportamento intermediário. O $\overline{\text{IAE}}$ cresce de forma progressiva com o aumento do setpoint, porém mantém fraca dependência de K e M para valores moderados de controladores. Mesmo em regimes mais exigentes (setpoints de 120 e 150), o $\overline{\text{IAE}}$ não apresenta regiões críticas associadas ao compartilhamento agressivo. Em conjunto, esses resultados indicam que a qualidade média de controle é fortemente dependente do regime operacional e da dinâmica da planta, sendo relativamente pouco afetada pelo número de controladores, desde que K não seja extremamente pequeno.



(a) Motor DC **(b) Tanque.**

(c) Veículo.

Figure 2. Mapas de calor do $\overline{\text{IAE}}$ no espaço (M, K) para as plantas, considerando os setpoints avaliados.

3.2. Tráfego de Rede

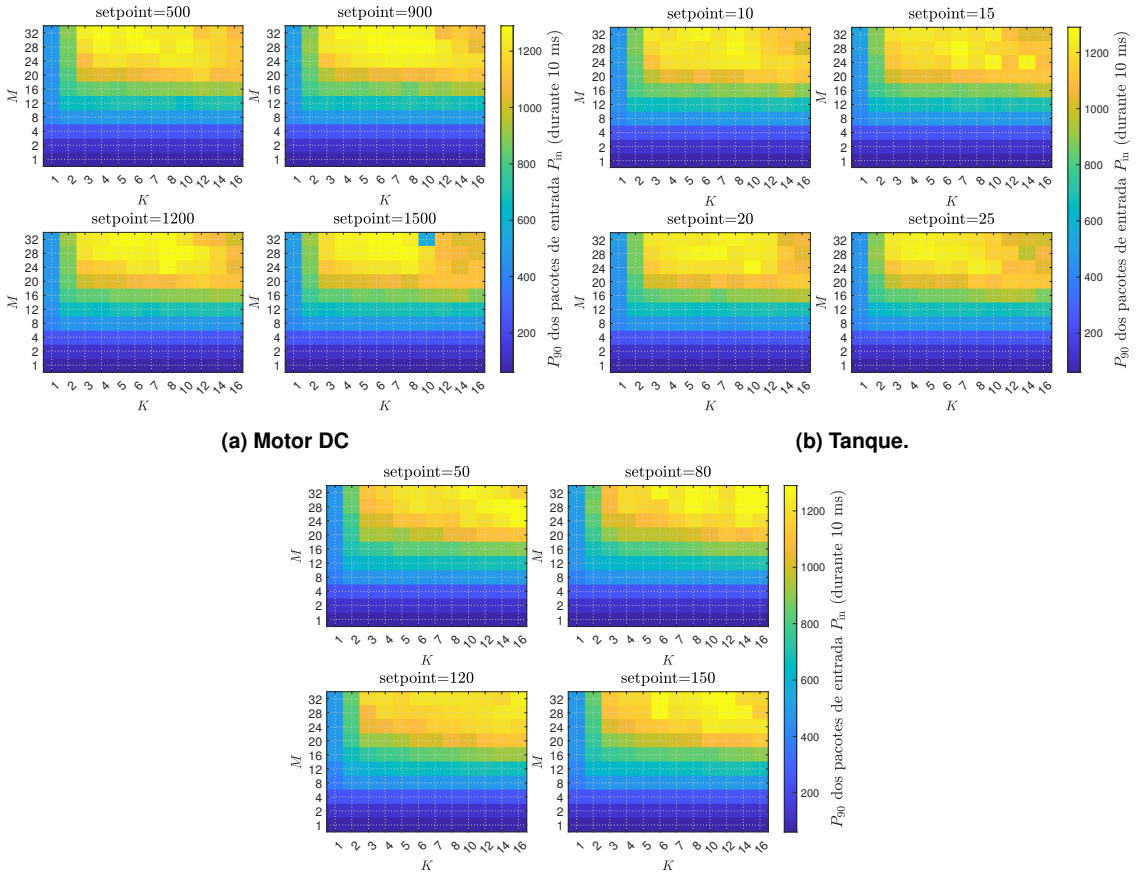
As Figuras 3a, 3b e 3c mostram os mapas de calor do P_{90} do número de pacotes de entrada, P_{in}, por janela de 10 ms para as três plantas.

Para todas as plantas, observa-se que o tráfego cresce de forma monotônica com M, refletindo diretamente o aumento do número de dispositivos IIoT ativos. A influência de K sobre o volume agregado de tráfego é secundária, estando associada principalmente à distribuição da carga entre os controladores. No motor DC e no tanque, o padrão de tráfego é altamente regular e praticamente independente do setpoint, caracterizando o tráfego como um fator exógeno de carga. No veículo, o comportamento é semelhante, com crescimento suave e previsível do tráfego à medida que M aumenta. Esses resultados reforçam que o tráfego de rede, embora essencial para explicar a carga computacional e os atrasos observados, não é o principal fator diferenciador entre plantas.

3.3. Uso de CPU

As Figuras 4a, 4b e 4c apresentam os mapas de calor do percentil 90 (P_{90}) associado ao uso de CPU, u_{CPU}, (em $\%$) em função de (M, K).

Para as três plantas, observa-se saturação severa do valor de u_{CPU} para valores baixos de K, especialmente quando M é elevado. À medida que K aumenta, o P_{90} de



(a) Motor DC

(b) Tanque.

(c) Veículo.

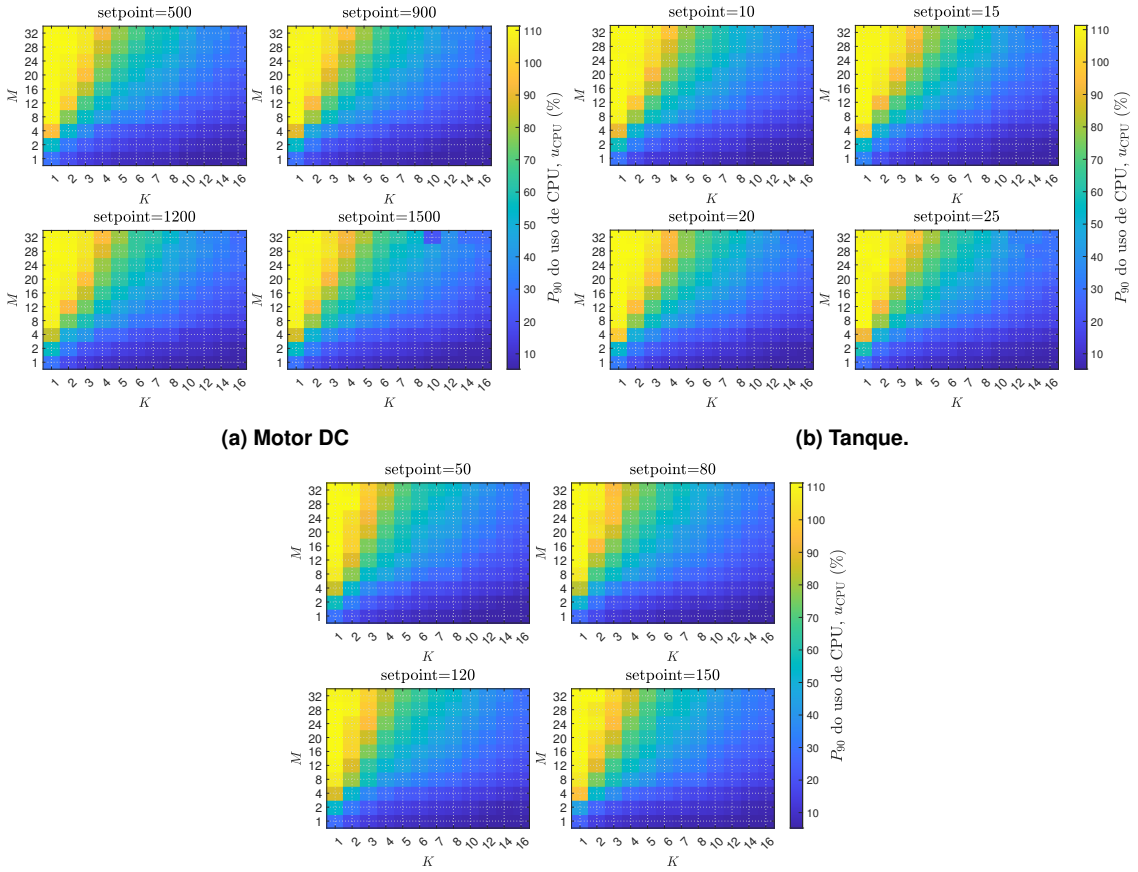
Figure 3. Mapas de calor do P_{90} de P_{in} no espaço (M, K) para as plantas, considerando os setpoints avaliados.

u_{CPU} decresce quase monotonicamente, evidenciando a redução da contenção computacional nos pods de inferência. O motor DC apresenta a maior pressão computacional, com valores P_{90} associados u_{CPU} mais elevados em todos os setpoints, refletindo uma inferência mais frequente e sensível ao regime. O tanque apresenta comportamento semelhante, porém com níveis absolutos ligeiramente menores. Já o veículo é a planta menos exigente do ponto de vista computacional, permitindo compartilhamento mais agressivo sem saturação. Importante destacar que o aumento do setpoint desloca os níveis absolutos de u_{CPU}, mas não altera a estrutura da superfície (M, K), indicando que o regime operacional atua como um fator de amplificação da carga, e não como um fator estrutural.

3.4. Requisitos de Tempo Real

As Figuras 5a, 5b e 5c apresentam os mapas de calor do P_{90} do tempo de loop de controle (*closed-loop control*), expresso aqui como Δ_{clc} e dado em ms. Enquanto as Figuras 6a, 6b e 6c mostram o P_{90} do atraso de requisição do IIoT, expresso neste artigo como τ_r em ms.

Para o motor DC, as métricas temporais, Δ_{clc} e τ_r revelam claramente regiões de operação inviáveis para valores muito baixos de K, especialmente em setpoints elevados. O valor de P_{90} para Δ_{clc} cresce rapidamente com M quando $K \leq 2$, atingindo valores incompatíveis com aplicações de tempo real. A transição para um regime estável ocorre



(a) Motor DC **(b) Tanque.**

(c) Veículo.

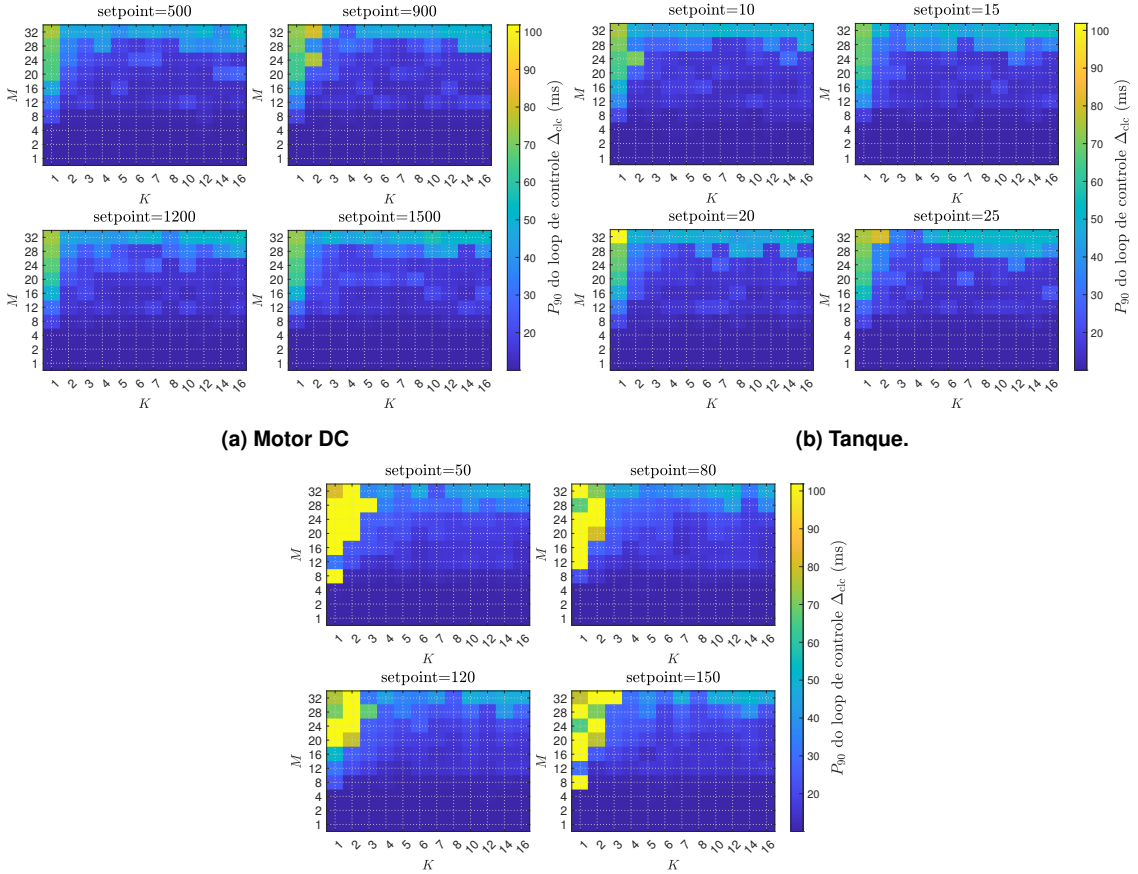
Figure 4. Mapas de calor do P_{90} do u_{CPU} no espaço (M, K) para as plantas, considerando os setpoints avaliados.

tipicamente entre $K \approx 4$ e $K \approx 6$, dependendo do setpoint. O tanque apresenta maior robustez temporal. Embora haja aumento de Δ_{clc} e de τ_r para K muito pequenos e M elevados, as regiões críticas são menos extensas do que no motor DC. Para a maioria dos cenários, valores de $K \geq 3$–4 são suficientes para manter o Δ_{clc} em níveis aceitáveis. A planta do veículo é a mais tolerante do ponto de vista temporal. Mesmo em regimes de compartilhamento agressivo, os valores de Δ_{clc} e τ_r permanecem baixos para a maior parte do espaço (M, K). Regiões críticas aparecem apenas para setpoints elevados combinados com $K = 1$ e grandes valores de M.

3.5. Sumarização dos Resultados Experimentais

A análise conjunta dos resultados permite destacar quatro conclusões fundamentais:

- Existe uma região consistente de operação com $K < M$ para todas as plantas analisadas, na qual a qualidade de controle, $\overline{\mathrm{IAE}}$, e os requisitos de tempo real (Δ_{clc} e τ_r) são atendidos simultaneamente.
- O valor mínimo de K necessário para operação estável depende fortemente da dinâmica da planta e do regime operacional (setpoint).
- O valores $\overline{\mathrm{IAE}}$ são insuficientes para caracterizar os limites do sistema. Todavia, as métricas de P_{90} de (Δ_{clc} e τ_r são essenciais para revelar violações de tempo real.



(a) Motor DC **(b) Tanque.**

(c) Veículo.

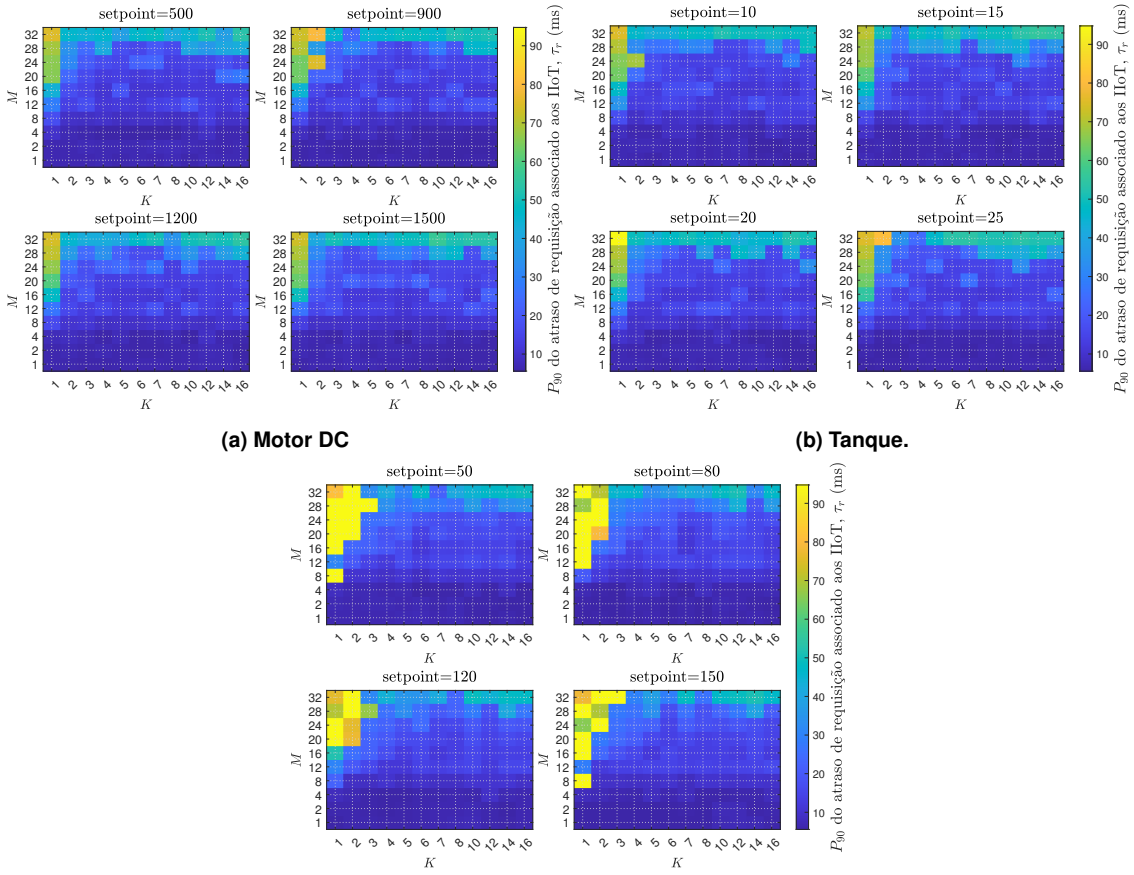
Figure 5. Mapas de calor do P_{90} do Δ_{clc} no espaço (M, K) para as plantas, considerando os setpoints avaliados.

- O compartilhamento de controladores é particularmente eficaz para plantas com maior inércia física, como o tanque e o veículo, enquanto plantas mais sensíveis, como o motor DC, exigem valores maiores de K.

4. Política de Dimensionamento Control-Aware

Os resultados apresentados na Seção 3 demonstram que o desempenho de sistemas de controle compartilhado em ambientes de borda não pode ser adequadamente gerenciado por políticas de escalonamento baseadas apenas em métricas tradicionais de computação, como uso médio de CPU. Em particular, observou-se que métricas de QoC e, principalmente, o P_{90} de métricas temporais como Δ_{clc} e τ_r revelam regiões de operação inviáveis que não são capturadas por abordagens convencionais de escalonamento horizontal. Com base nesses achados, este trabalho propõe uma política de dimensionamento *control-aware*, cujo objetivo é determinar dinamicamente o número mínimo de controladores compartilhados (K) necessário para garantir simultaneamente estabilidade temporal, qualidade de controle aceitável e eficiência computacional. A política proposta fundamenta-se em três princípios centrais:

1. Consciência da aplicação de controle: o dimensionamento deve considerar explicitamente a dinâmica da planta e o regime operacional (setpoint), e não apenas a carga computacional agregada.



(a) Motor DC **(b) Tanque.**

(c) Veículo.

Figure 6. Mapas de calor do P_{90} do τ_r no espaço (M, K) para as plantas, considerando os setpoints avaliados.

2. Garantia de tempo real: a estabilidade temporal do sistema é avaliada por métricas de percentil P_{90}, de modo a capturar violações raras, porém críticas, que impactam o fechamento da malha de controle.
3. Minimização de recursos: sempre que possível, a política seleciona valores de K estritamente menores que M, evitando superprovisionamento e desperdício de recursos.

Esses princípios refletem diretamente os padrões observados nas três plantas analisadas, nos quais a relação ótima entre K e M depende fortemente da dinâmica física do sistema controlado. Assim, pode-se definir valor mínimo de controladores, $K_{\min}(p, s)$, necessários para um determinado cenário como:

$$K_{\min}(p,s) = \min_{K} \quad \text{s.t.} \quad \begin{cases} \overline{\text{IAE}}(p,s,M,K) \leq \delta_{\overline{\text{IAE}}}(p,s) \\ \Delta_{\text{clc}}(p,s,M,K) \leq \Delta_{\text{clc}}^{\max} \end{cases} \qquad \forall M \in \mathcal{M}, \qquad (1)$$

onde p representa a planta (motor DC, tanque ou veículo), s representa o valor setpoint, $\delta_{\overline{\text{IAE}}}$ é um limiar aceitável de degradação da QoC (definido experimentalmente), $\Delta_{\text{clc}}^{\max}$ é o limite máximo admissível para o P_{90} de Δ_{clc} e $\mathcal{M}$ é o conjunto de cargas de dispositivos IIoT consideradas. Na prática, conforme observado no experimento, a restrição temporal é dominante, uma vez que o IAE permanece aceitável em ampla faixa do espaço (M, K), enquanto violações de tempo real surgem de forma abrupta para valores baixos de K.

A política de dimensionamento control-aware opera de forma determinística e pode ser implementada tanto de maneira offline (planejamento) quanto online (em conjunto com mecanismos de escalonamento do K8s). O Algoritmo 1 descreve o procedimento básico para determinação de $K_{\min}$. O algoritmo seleciona o menor valor de K que satisfaz simultaneamente os requisitos de QoC e tempo real para todas as cargas consideradas. Caso nenhum valor intermediário atenda às restrições, retorna-se o valor máximo disponível de controladores.

Algoritmo 1 Determinação de $K_{\min}$

Require: Planta p, setpoint s, conjunto de cargas $\mathcal{M}$, limiares $\delta_{\overline{\mathrm{IAE}}}$, $\Delta_{\mathrm{clc}}^{\max}$
for cada K em ordem crescente **do**
  if para todo $M \in \mathcal{M}$: **then**
    $\overline{\mathrm{IAE}}(p, s, M, K) \leq \delta_{\overline{\mathrm{IAE}}}(p, s)$ **e** $\Delta_{\mathrm{clc}}(p, s, M, K) \leq \Delta_{\mathrm{clc}}^{\max}$
    return K
  end if
end for
return $K_{\max}$

A Tabela 1 resume os valores mínimos de controladores compartilhados necessários para satisfazer os requisitos temporais, considerando três critérios: mediana, P_{90} e pior caso ao longo das cargas avaliadas. Observa-se que o critério de pior caso resulta em valores excessivamente conservadores, enquanto o critério baseado no P_{90} fornece um compromisso mais adequado entre garantia de tempo real e eficiência de recursos. Em particular, plantas com maior inércia física, como o tanque e o veículo, requerem valores significativamente menores de $K_{\min}$ quando comparadas ao motor DC, evidenciando que políticas de dimensionamento devem ser dependentes da planta e do regime operacional. Esses resultados confirmam que $K \ll M$ é suficiente em ampla faixa de cenários, tornando o controle compartilhado em borda viável e eficiente.

Table 1. Valores mínimos de controladores compartilhados ($K_{\min}$) por planta e setpoint, considerando $\Delta_{\mathrm{clc}}^{\max} = 40$ ms.

Planta	Setpoint	$K_{\min}^{\mathrm{median}}$	$K_{\min}^{p90}$	$K_{\min}^{\mathrm{worst}}$
Motor DC	500 rpm	2	4	5
	900 rpm	2	4	4
	1200 rpm	2	5	8
	1500 rpm	1	2	2
Tanque	10 m	1	3	3
	15 m	2	3	3
	20 m	1	2	2
	25 m	2	3	3
Veículo	50 km/h	2	4	4
	80 km/h	2	4	5
	120 km/h	2	4	4
	150 km/h	2	4	4

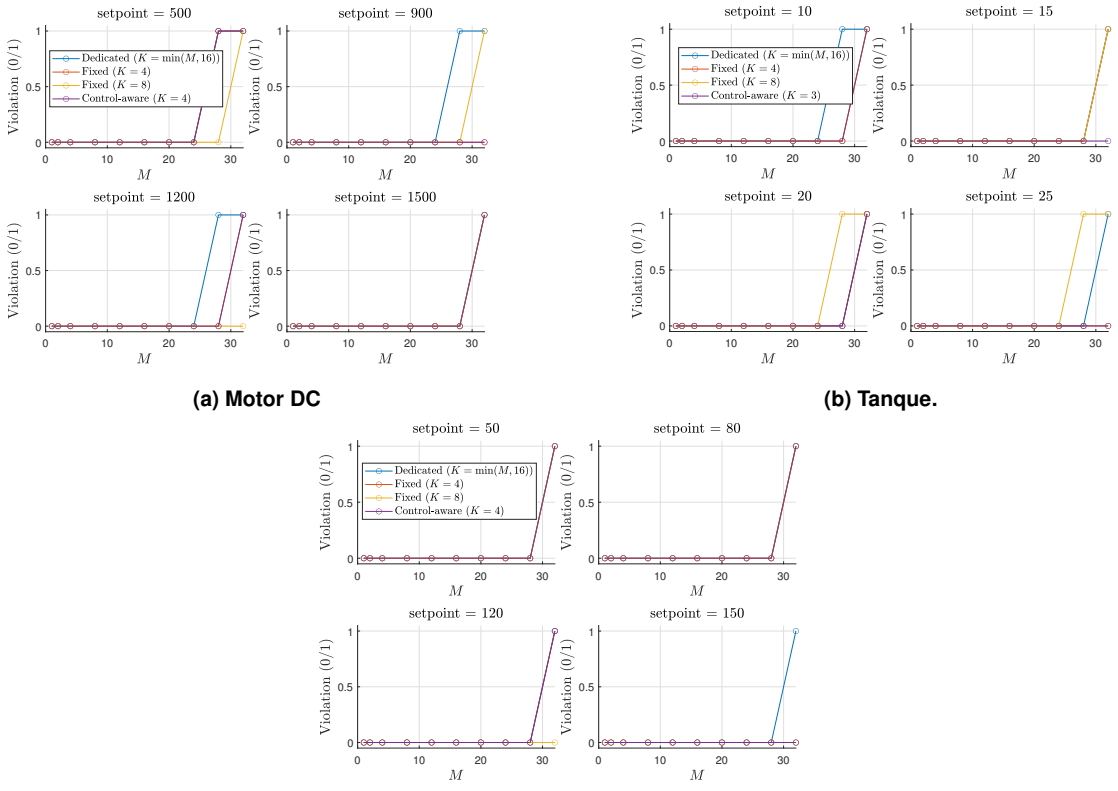
5. Avaliação da Política de Dimensionamento

Foram comparadas três classes de estratégias. No provisionamento dedicado, cada dispositivo IIoT é atendido por um controlador exclusivo, resultando em $K = \min(M, 16)$, o que representa o limite superior em termos de consumo de recursos e serve como referência de desempenho temporal. As estratégias de provisionamento fixo mantêm o

número de controladores constante, independentemente da carga, sendo avaliados valores típicos da prática, especificamente $K = 4$ e $K = 8$. A política control-aware, por sua vez, define o número de controladores com base no valor $K_{\min}^{p90}$ obtido experimentalmente para cada combinação de planta e setpoint (Tabela 1), mantendo esse valor constante ao longo da variação de M.

O critério principal de avaliação é a estabilidade temporal do sistema de controle, expressa pelo P_{90} do tempo de loop. Considera-se que ocorre uma violação temporal quando o P_{90} do valor de Δ_{clc} excede 40 ms, limiar adotado para capturar eventos raros, porém críticos, que comprometem o fechamento da malha de controle. Como métrica complementar, analisa-se o P_{90} do u_{CPU}, representando o custo computacional associado a cada estratégia.

As ocorrências de violações temporais em função do número de dispositivos IIoT são apresentadas na Fig. 7. Para a planta do motor DC (Fig. 7a), observa-se que as estratégias de provisionamento fixo apresentam falhas abruptas à medida que M aumenta, especialmente em regimes operacionais mais exigentes. O provisionamento dedicado elimina violações na maior parte do espaço avaliado, porém ao custo de um número elevado de controladores ativos. Em contraste, a política control-aware mantém o P_{90} do valor de Δ_{clc} abaixo do limiar estabelecido em uma faixa significativamente maior de valores de M, utilizando menos controladores do que o caso dedicado.



(a) Motor DC

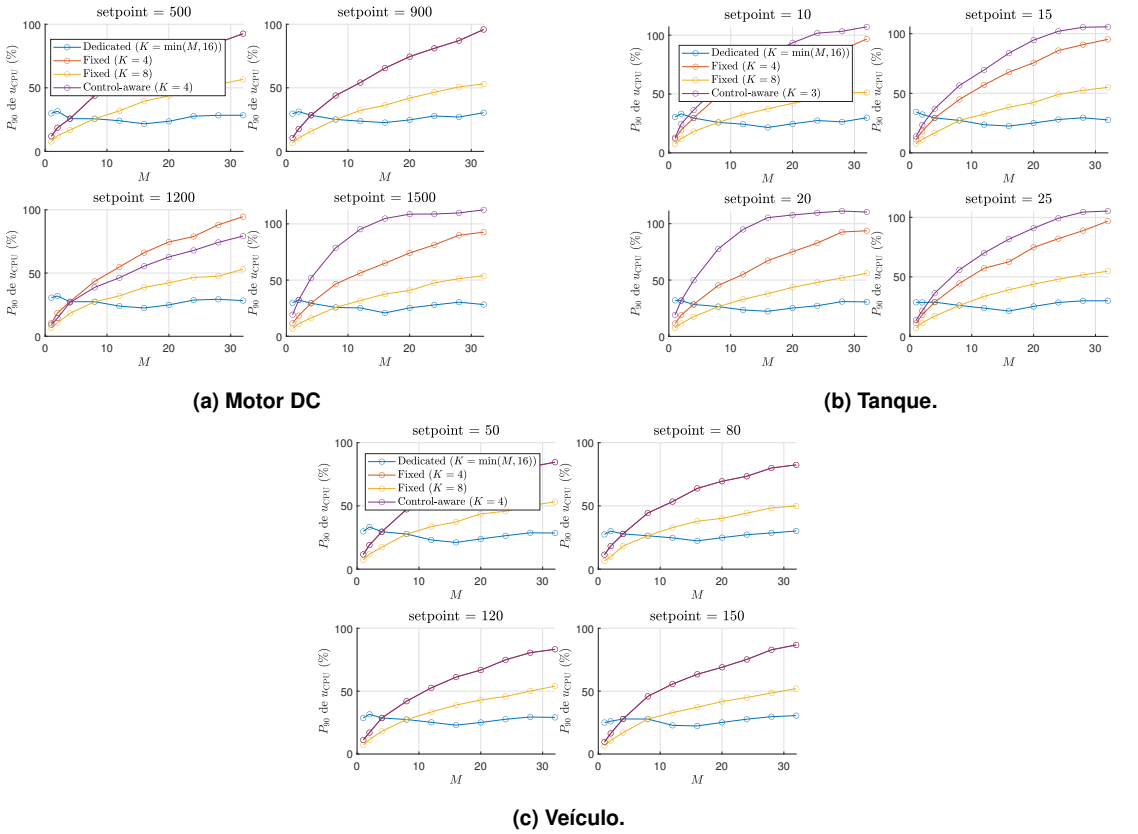
(b) Tanque.

(c) Veículo.

Figure 7. Ocorrência de violações temporais em função do número de dispositivos IIoT (M) para as plantas considerando diferentes estratégias de provisionamento.

Para a planta do tanque (Fig. 7b), cuja dinâmica é lenta e naturalmente amortecida, as violações temporais são raras e restritas a cenários extremos quando estratégias fixas são utilizadas. A política control-aware explora essa robustez, mantendo estabilidade temporal em praticamente todo o espaço avaliado com valores reduzidos de K. No caso do veículo (Fig. 7c), observa-se o comportamento mais tolerante entre as plantas analisadas, com a política proposta garantindo estabilidade temporal mesmo em regimes de compartilhamento agressivo.

O custo computacional associado às estratégias é analisado por meio do P_{90} do u_{CPU}, apresentado na Fig. 8. Para as três plantas (Figs. 8a–8c), o provisionamento dedicado mantém níveis moderados de CPU ao custo de um crescimento linear no número de controladores, enquanto as estratégias fixas apresentam aumento significativo do uso de CPU à medida que M cresce, mesmo quando ainda ocorrem violações temporais. A política control-aware posiciona-se de forma intermediária, apresentando uso de CPU sistematicamente menor do que as estratégias fixas e evitando o superprovisionamento característico do caso dedicado.



(a) Motor DC

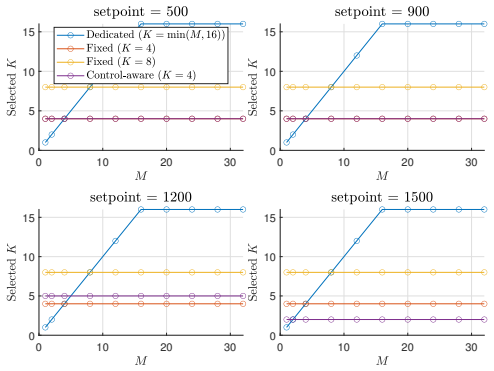
(b) Tanque.

(c) Veículo.

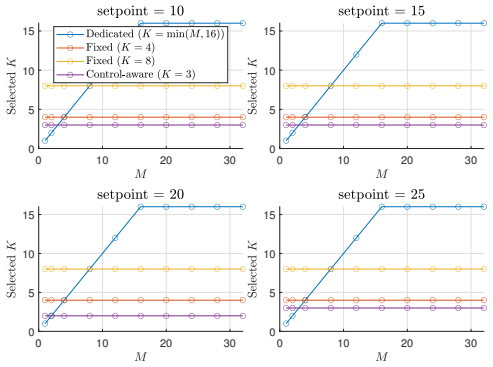
Figure 8. P_{90} do u_{CPU} em função do número de dispositivos IIoT (M) para as plantas considerando diferentes estratégias de provisionamento.

As decisões de escalonamento adotadas por cada estratégia são explicitadas na Fig. 9. Enquanto o provisionamento dedicado aumenta K proporcionalmente a M (Figs. 9a–9c) e as estratégias fixas mantêm valores constantes independentemente do regime, a política control-aware seleciona valores mínimos e constantes de K para cada planta e setpoint. Essa característica explica diretamente o comportamento observado

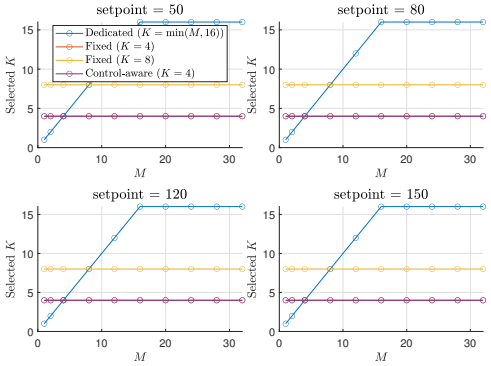
nas métricas temporais e de CPU, evidenciando que a estabilidade do sistema pode ser garantida sem a necessidade de K crescer linearmente com a carga.



(a) Motor DC



(b) Tanque.



(c) Veículo.

Figure 9. Decisões de escalonamento do número de controladores (K) em função do número de dispositivos IIoT (M) para as plantas considerando diferentes estratégias de provisionamento.

A avaliação demonstra que políticas de dimensionamento baseadas apenas em valores fixos de K são incapazes de se adaptar adequadamente às variações de carga e regime operacional, enquanto o provisionamento dedicado resulta em uso excessivo de recursos. A política control-aware proposta alcança um compromisso superior entre garantia de tempo real e eficiência computacional, mantendo $K \ll M$ em ampla faixa de cenários e tornando o controle compartilhado em ambientes de borda uma alternativa prática e eficiente.

6. Conclusão

Este trabalho investigou o dimensionamento de controladores fuzzy compartilhados em ambientes de borda baseados em K8s no contexto de sistemas IIoT com controle em tempo real. A partir de uma análise experimental sistemática envolvendo múltiplas plantas, regimes operacionais e configurações de carga, foi possível caracterizar o compromisso entre QoC, requisitos temporais e custo computacional no espaço (M, K). Os resultados demonstraram que métricas tradicionais de computação são insuficientes para capturar limites críticos do sistema, sendo as métricas temporais de percentil fundamentais para revelar regiões de operação inviáveis. Com base nesses achados, foi proposta e avaliada uma política de dimensionamento *control-aware*, capaz de selecionar o número

mínimo de controladores que satisfaz simultaneamente restrições de QoC e tempo real, superando estratégias de provisionamento fixo e dedicado em termos de eficiência e robustez. Esses resultados reforçam a viabilidade do controle compartilhado em borda quando o dimensionamento é guiado por métricas sensíveis à aplicação de controle.

Agradecimentos

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio e financiamento. Esta pesquisa integra o Instituto Nacional de Ciência e Tecnologia em Redes de Comunicação e Internet das Coisas Inteligente (ICoNIoT), financiado pelo CNPq (processo nº 405940/2022-0) e pela CAPES (Código de Financiamento nº 88887.954253/2024-00).

References

Caiazza, G., Lisovenko, T., Ferrara, P., Berti, F., Ferrari, F., Zaupa, A., and Zhang, G. (2025). From legacy to intelligent iiot systems: Automation, scalability and elasticity. In *2025 IEEE 22nd International Conference on Software Architecture (ICSA)*, pages 255–266. IEEE.

Fernandes, M. (2026). Experimental dataset for control-aware autoscaling of shared controllers in edge kubernetes systems. V1. DOI: 10.17632/b8r6rpv8ch.1.

Hall, J., Morrow, B., and Godbehere, A. (2024). Enhancing iiot infrastructures with kubernetes: Advanced edge cluster management. In *2024 11th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, pages 132–139. IEEE.

Kaur, K., Garg, S., Kaddoum, G., Ahmed, S. H., and Atiquzzaman, M. (2020). Keids: Kubernetes-based energy and interference driven scheduler for industrial iot in edge-cloud ecosystem. *IEEE Internet of Things Journal*, 7(5):4228–4237.

Kuchuk, H. and Malokhvii, E. (2024). Integration of iot with cloud, fog, and edge computing: a review. *Advanced Information Systems*, 8(2):65–78.

Li, D. C., Huang, C.-T., Tseng, C.-W., and Chou, L.-D. (2021). Fuzzy-based microservice resource management platform for edge computing in the internet of things. *Sensors*, 21(11).

Sadikhov, S., Tärneberg, W., Fitzgerald, E., Peng, H., and Nyberg, C. (2025). Aoa and aoi in modern industrial control. In *2025 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 01–07. IEEE.

Silva, S. N., Goldbarg, M. A. S. d. S., Silva, L. M. D. d., and Fernandes, M. A. C. (2024). Application of fuzzy logic for horizontal scaling in kubernetes environments within the context of edge computing. *Future Internet*, 16(9).

Silva, S. N., Torquato, M. F., and Fernandes, M. A. (2019). Comparison of binary and fuzzy logic in feedback control of dynamic systems. *International Journal of Dynamics and Control*, 7(3):1056–1064.

Yan, C., Xia, Y., Yang, H., and Zhan, Y. (2024). Cloud control for iiot in a cloud-edge environment. *Journal of Systems Engineering and Electronics*, 35(4):1013–1027.