

## CONTROLADOR PREDITIVO DE HORIZONTE INFINITO EMBARCADO: APLICAÇÃO *HARDWARE-IN-THE-LOOP* A UM SISTEMA DE BOMBEIO CENTRÍFUGO SUBMERSO

BRUNO A. SANTANA,\* RAONY M. FONTES,† MÁRCIO A. F. MARTINS\*

\*Programa de Pós-Graduação em Mecatrônica, Escola Politécnica, Universidade Federal da Bahia, Salvador (BA), 40210630.

†Departamento de Engenharia Química, Escola Politécnica, Universidade Federal da Bahia, Salvador (BA), 40210630.

Emails: bruno.aguiar@ufba.br, raony@ufba.br, marciomartins@ufba.br

**Abstract**— This work presents the implementation of an infinite horizon model predictive controller applied to an ESP-lifted oil well system. The practical aspects of implementation are investigated through the hardware-in-the-loop test in a microcontroller programmed in C/C++. Suitable use of an efficient active-set-type optimization algorithm is proposed to solve the quadratic programming problem associated with the control law, making the approach implementable in practice and resulting in computation times in the order of milliseconds. Furthermore, the implemented zone control strategy contributes to accommodate the dynamic operational envelope-type path constraints, thus preserving controller feasibility.

**Keywords**— Predictive Control, Electric Submersible Pump (ESP), Embedded Systems, Optimization.

**Resumo**— Este trabalho apresenta a implementação de um controlador preditivo de horizonte infinito aplicado a um sistema de poço de petróleo elevado por BCS. Os aspectos práticos de implementação são investigados através do teste *hardware-in-the-loop* em um microcontrolador programado em C/C++. O uso adequado de um eficiente algoritmo de otimização do tipo conjuntos ativos é proposto para solucionar o problema de programação quadrática associado à lei de controle, tornando a abordagem implementável na prática e resultando em tempos de cômputo na ordem de milissegundos. Além disso, a estratégia de controle por faixa implementada contribui para acomodar as restrições dinâmicas do envelope operacional do BCS, preservando assim a factibilidade do controlador.

**Palavras-chave**— Controle Preditivo, Bombeio Centrífugo Submerso (BCS), Sistemas Embarcados, Otimização.

### 1 Introdução

O controle preditivo baseado em modelo (MPC) é uma estratégia de controle ótimo conhecida por tratar sistemas multivariáveis sujeitos a restrições e fornecer uma solução sistemática para problemas de controle dessa natureza. Além disso, objetivos econômicos podem ser facilmente incluídos na formulação de um MPC, característica que o torna uma das técnicas de controle mais difundidas na indústria de processos (Camacho and Bordons, 2004). No âmbito da indústria de petróleo, o MPC corresponde à tecnologia padrão usada no segmento *downstream*, sobretudo no processo de refino do petróleo (Binder et al., 2014).

Por outro lado, no segmento *upstream*, o qual compreende as atividades de exploração e produção de petróleo, o uso de técnicas de controle automático ainda é uma questão incipiente, sobretudo em campos de produção localizados no mar (*offshore*), onde geralmente o monitoramento e o controle das variáveis de processo são realizados de forma manual. Esse é o caso, por exemplo, dos campos de produção de petróleo nos quais utiliza-se o método de elevação artificial do tipo bombeio centrífugo submerso (BCS). Tal método é composto por uma bomba centrífuga multiestágio instalada no fundo do poço para aumentar a pressão do fluido e viabilizar sua elevação até a superfície, sendo usualmente empregado quando se

deseja produzir volumes significativos de petróleo (Liang et al., 2015), desde que o reservatório não possua uma alta razão gás-óleo.

A operação de um sistema BCS consiste no ajuste da frequência rotacional da bomba e da abertura da válvula de produção (*choke*) situada na cabeça no poço. Além disso, um conjunto de restrições do poço e da bomba deve ser satisfeito, de tal forma que o sistema opere dentro das condições recomendadas pelos fabricantes dos equipamentos que compõem o sistema. O conjunto de restrições impostas pelas forças axiais que atuam na bomba (*downthrust* e *upthrust*) dão origem ao chamado envelope operacional (Takacs, 2009), o qual delimita uma região de operação segura e estável. A operação fora desse envelope pode ocasionar desgates mecânicos severos na bomba, reduzindo a vida útil do equipamento ou mesmo levando à parada do processo para manutenção, o que traz consigo perdas econômicas significativas (Pavlov et al., 2014). No entanto, a operação próxima das restrições pode ser requerida para maximizar a produção ou minimizar o consumo de energia, o que exige técnicas de controle automático que possam reduzir a variabilidade das variáveis de processo, atributo característico de um controlador MPC.

No contexto da comunidade científica, esforços de pesquisa recentes têm sido direcionados para o desenvolvimento e implementação de con-

troladores MPC em sistemas BCS, com o uso de diferentes estratégias para contemplar as restrições do processo e atender aos objetivos de controle e econômicos. O trabalho de Pavlov et al. (2014) foi um dos precursores no segmento supracitado, apresentando a implementação de um controlador MPC em uma planta piloto do sistema BCS. Na formulação do controlador, o objetivo de controle consiste no rastreamento da pressão de *intake* e as restrições do envelope operacional são incorporadas explicitamente. Além disso, os autores contemplam o objetivo econômico ao definirem um *target* para a abertura da válvula *choke*, a fim de minimizar o consumo de energia do BCS. Seguindo essa mesma abordagem, Binder et al. (2014) investigaram os aspectos de implementação do MPC em um controlador lógico programável, realizando simulações *hardware-in-the-loop* com a planta virtual. No entanto, o consumo de energia foi minimizado através da regulação da corrente do motor e as restrições do envelope operacional foram contempladas pelo rastreamento da taxa de vazão do BCS. Em trabalhos posteriores, outros autores propuseram extensões da formulação MPC explorada por Pavlov et al. (2014), como pode ser visto em (Krishnamoorthy et al., 2016; Binder et al., 2019; Delou et al., 2019).

Uma questão em aberto nos trabalhos supracitados, diz respeito à garantia de estabilidade em malha fechada e factibilidade dos controladores MPC propostos, visto que tais formulações são baseadas em horizontes de predição finitos e suas leis de controle podem se tornar facilmente inviáveis devido ao conflito típico que ocorre entre as restrições do sistema BCS. Diante disso, Fontes et al. (2020) propuseram a aplicação de um MPC de horizonte de predição infinito (IHMP) para um sistema BCS, considerando o modelo dinâmico concebido por Pavlov et al. (2014). Os autores utilizaram uma estratégia de controle por faixa para rastrear o envelope operacional (restrições variantes no tempo), que suaviza, somente quando necessário, o conflito típico existente nas restrições das saídas do sistema BCS, através do uso de restrições terminais suavizadas e de *set-points* artificiais. Com isso, as propriedades relacionadas à estabilidade e à factibilidade da lei de controle do IHMP são preservadas, tornando-a adequada para aplicações em tempo real. Além disso, as restrições de *downthrust* e *upthrust* são consideradas explicitamente no problema de otimização e o *target* econômico relacionado à abertura da válvula *choke* é usado a fim de reduzir o consumo de energia do sistema.

Este artigo tem como proposta investigar os aspectos de implementação do controlador IHMP explorado em Fontes et al. (2020), validando-o através do teste *hardware-in-the-loop* com a planta virtual simulada em MATLAB. A viabilidade de embarcar o IHMP é investigada

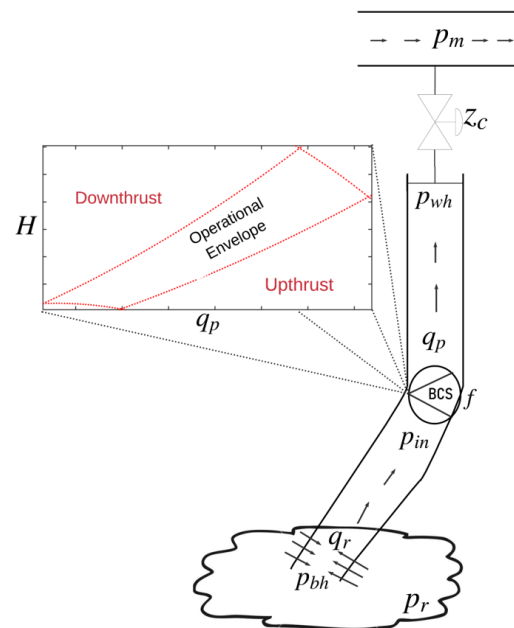


Figura 1: Esquema de um processo de produção de petróleo com instalação BCS.

utilizando um microcontrolador programado com a linguagem C/C++, no qual um *solver* de otimização para programação quadrática, baseado no método dos conjuntos ativos, também foi implementado para calcular a lei de controle a cada instante de amostragem.

## 2 Descrição do BCS e da formulação IHMP

### 2.1 Sistema BCS

A Figura 1 descreve um sistema típico de poço de petróleo elevado por BCS, destacando suas principais variáveis operacionais, bem como condições operacionais limitantes do envelope operacional do BCS. As condições de *downthrust* e *upthrust* são variantes no tempo, o que torna a operação do sistema uma tarefa desafiadora, cujo uso de estratégias de controle multivariável do tipo MPC pode auxiliar neste cenário crítico. O petróleo do reservatório passa pelo BCS, que fornece pressão adicional ao fluido, permitindo o fluxo até as instalações de superfície. A operação de um poço de petróleo elevado por BCS geralmente tem dois graus de liberdade, ou seja, a frequência rotacional da bomba ( $f$ ) e a abertura da válvula *choke* de produção ( $z_c$ ).

O modelo dinâmico de terceira ordem proposto por Pavlov et al. (2014), incluindo seus valores de parâmetros e suposições, será usado para a simulação dinâmica do poço de petróleo elevado por BCS. O modelo descrito por um conjunto de equações diferenciais é resumido como segue.

$$\begin{cases} \dot{p}_{wh} = 1.54 \times 10^8 (q_p - q_c) \\ \dot{p}_{bh} = 0.8584(p_r - p_{bh}) - 3.7 \times 10^8 q_p \\ \dot{q}_p = 5.02 \times 10^{-9} [p_{bh} - p_{wh} - 6.30 \times 10^8 q_p^{1.75} \\ \quad + 9.32 \times 10^3 (H - 1 \times 10^3)] \\ q_c = 2 \times 10^{-3} z_c \sqrt{p_{wh} - p_m} \\ p_{in} = p_{bh} - 1.85 \times 10^8 q_p^{1.75} - 1.9 \times 10^6 \\ H = 0.2664f^2 + 133.09fq_p - 1.41 \times 10^6 q_p^2 \end{cases} \quad (1)$$

onde  $p_{wh}$ ,  $p_{bh}$ ,  $p_{in}$ ,  $p_m$  e  $p_r$  são as pressões da cabeça do poço, de fundo, de *intake*, de *manifold* e do reservatório, respectivamente;  $q_p$  e  $q_c$  são a taxa de vazão média da coluna de produção e a taxa de vazão na *choke* de produção, respectivamente; e  $H$  é o *head* da bomba.

## 2.2 Lei de controle IHMPC

A lei de controle aqui avaliada para o controle do sistema BCS é baseada em uma formulação MPC de horizonte infinito, que faz uso de um modelo orientado à predição da saída (OPOM, do inglês *Output Predicted-Oriented Model*).

OPOM é um modelo em espaço de estados linear construído a partir de uma expressão analítica da resposta ao degrau do sistema, em símbolos (González and Odloak, 2009):

$$\begin{aligned} \begin{bmatrix} \mathbf{x}^s(k+1) \\ \mathbf{x}^{st}(k+1) \end{bmatrix} &= \underbrace{\begin{bmatrix} \mathbf{I}_{ny} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}^{st} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \mathbf{x}^s(k) \\ \mathbf{x}^{st}(k) \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{B}^s \\ \mathbf{B}^{st} \end{bmatrix}}_{\mathbf{B}} \Delta \mathbf{u}(k), \\ \mathbf{y}(k) &= \underbrace{\begin{bmatrix} \mathbf{I}_{ny} & \Psi \end{bmatrix}}_{\mathbf{C}} \begin{bmatrix} \mathbf{x}^s(k) \\ \mathbf{x}^{st}(k) \end{bmatrix}. \end{aligned} \quad (2)$$

Na equação de estado definida em (2), o componente de estado  $\mathbf{x}^s$  corresponde aos polos integradores produzidos pela forma incremental das entradas, e  $\mathbf{x}^{st}$  corresponde aos modos estáveis do sistema.  $\mathbf{F}^{st}$  está relacionada aos polos do sistema, enquanto  $\mathbf{B}^s$  e  $\mathbf{B}^{st}$  correspondem aos coeficientes da expansão em frações parciais; mais detalhes podem ser encontrados em (Odloak, 2004).

A lei de controle do IHMPC visa solucionar o seguinte problema de otimização a cada instante de amostragem:

### Problema 1

$$\begin{aligned} &\min_{\Delta \mathbf{u}_k, \mathbf{y}_{sp,k}, \delta_{\mathbf{y},k}, \delta_{\mathbf{u},k}} V_k, \\ V_k &= \sum_{j=1}^m \|\mathbf{y}(k+j|k) - \mathbf{y}_{sp,k} - \delta_{\mathbf{y},k}\|_{\mathbf{Q}_y}^2 + \\ &\quad \sum_{j=0}^{m-1} \|\Delta \mathbf{u}(k+j|k)\|_{\mathbf{R}}^2 + \underbrace{\|\mathbf{x}^{st}(k+m|k)\|_{\mathbf{Q}_k}^2}_{T_{cost}} + \\ &\quad \sum_{j=0}^{m-1} \|\mathbf{u}(k+j|k) - \mathbf{u}_{tg} - \delta_{\mathbf{u},k}\|_{\mathbf{Q}_u}^2 + \\ &\quad \|\delta_{\mathbf{y},k}\|_{\mathbf{S}_y}^2 + \|\delta_{\mathbf{u},k}\|_{\mathbf{S}_u}^2, \end{aligned}$$

sujeito a (2) e:

$$\begin{cases} \mathbf{u}_{\min} \leq \mathbf{u}(k-1) + \sum_{i=0}^j \Delta \mathbf{u}(k+i|k) \leq \mathbf{u}_{\max} \\ -\Delta \mathbf{u}_{\max} \leq \Delta \mathbf{u}(k+j|k) \leq \Delta \mathbf{u}_{\max}, \quad j = 0, \dots, m-1 \\ \Delta \mathbf{u}(k+j|k) = \mathbf{0}, \quad \forall j \geq m, \end{cases} \quad (3)$$

$$\mathbf{y}_{\min}(k+j|k) \leq \mathbf{y}_{sp,k} \leq \mathbf{y}_{\max}(k+j|k), \quad j = 1, \dots, m, \quad (4)$$

$$\mathbf{x}^s(k+m|k) - \mathbf{y}_{sp,k} - \delta_{\mathbf{y},k} = \mathbf{0}, \quad (5)$$

$$\mathbf{u}(k+m-1|k) - \mathbf{u}_{tg} - \delta_{\mathbf{u},k} = \mathbf{0}, \quad (6)$$

em que  $m$  é o horizonte de controle,  $\Delta \mathbf{u}(k+j|k)$  são os incrementos das variáveis manipuladas e  $\mathbf{y}(k+j|k)$  são as predições das variáveis controladas no instante de tempo  $k+j$  dada a informação atual da planta no instante de tempo  $k$ ;  $\mathbf{y}_{sp,k}$  são *set-points* artificiais das variáveis controladas no esquema de controle por faixa;  $\mathbf{u}_{\max}$ ,  $\mathbf{u}_{\min}$ ,  $\Delta \mathbf{u}_{\max}$ ,  $\Delta \mathbf{u}_{\min}$ ,  $\mathbf{y}_{\max}$  e  $\mathbf{y}_{\min}$  são as restrições das variáveis manipuladas e seus incrementos, e das variáveis controladas, respectivamente;  $\Delta \mathbf{u}_k = [\Delta \mathbf{u}(k|k)^\top, \dots, \Delta \mathbf{u}(k+m-1|k)^\top]^\top$  é o vetor de ações de controle;  $\mathbf{Q}_y \in \mathcal{R}^{ny}$  e  $\mathbf{R} \in \mathcal{R}^{nu}$  são matrizes de ponderação das variáveis controladas ( $ny$ ) e manipuladas ( $nu$ ), respectivamente. Note que  $\Delta \mathbf{u}_k$ ,  $\mathbf{y}_{sp,k}$ ,  $\delta_{\mathbf{y}}$  e  $\delta_{\mathbf{u}}$  são as variáveis de decisão do problema de otimização, em particular,  $\delta_{\mathbf{y}}$  e  $\delta_{\mathbf{u}}$  são variáveis de folga a fim de garantir a factibilidade do controlador em cenários reais de desassociação modelo-planta,  $\mathbf{S}_u$  e  $\mathbf{S}_y$  são matrizes de ponderação das variáveis de folga;  $\mathbf{u}_{tg}$  são os *targets* das entradas e  $\mathbf{Q}_u$  é sua respectiva matriz de ponderação. Como o modelo de predição (2) possui modos integradores, restrições terminais ((5) e (6)) devem ser incluídas para evitar que o custo se torne ilimitado. Além disso, o custo terminal ( $T_{cost}$ ) é um ingrediente essencial a fim de tornar o MPC baseado em horizonte infinito implementável na prática, onde  $\bar{\mathbf{Q}}_k$  é a matriz de ponderação terminal calculada a partir da equação de Lyapunov do sistema, i.e.:

$$\bar{\mathbf{Q}}_k - (\mathbf{F}_k^{st})^\top \bar{\mathbf{Q}}_k \mathbf{F}_k^{st} = (\Psi^{st} \mathbf{F}_k^{st})^\top \mathbf{Q}_y \Psi^{st} \mathbf{F}_k^{st}.$$

O Problema 1 pode ser reescrito na forma compacta como o seguinte problema de programação quadrática (QP)

$$\min \left\{ \frac{1}{2} x^T H x + c^T x \mid Ax \leq b \right\} \quad (7)$$

onde  $x \in \mathbb{R}^n$  é o vetor de variáveis de decisão,  $c \in \mathbb{R}^n$  é o vetor gradiente,  $H \in \mathbb{S}_+^{n \times n}$  é a matriz Hessiana,  $A \in \mathbb{R}^{m \times n}$  e  $b \in \mathbb{R}^m$  são a matriz e o vetor de restrições, respectivamente. Note que as restrições terminais de igualdade ((5) e (6)) devem ser substituídas na função custo do Problema 1 para se obter a forma compacta. Com isso, a dimensão do problema de otimização resultante torna-se  $n = m.nu + ny$ .

### 3 Implementação embarcada

Nesta seção, os aspectos práticos de implementação do controlador serão descritos, incluindo o esquema do teste *hardware-in-the-loop* usado para a validação do algoritmo de controle embarcado no microcontrolador.

#### 3.1 Sistema embarcado

Para a implementação do controlador neste trabalho, utilizou-se como plataforma embarcada um microcontrolador ESP32. A série ESP32 emprega o microprocessador Tensilica Xtensa LX6 e pode operar a uma taxa de até 240 MHz, além de contar com memória RAM de 520 KB e memória *flash* de 4 MB. Apesar deste *hardware* não atender às exigências de robustez e aos protocolos industriais que um campo de produção de petróleo exige, seu uso justifica-se para validações dos algoritmos de controle através de testes com a planta virtual, sobretudo pelo seu baixo custo e alto poder de processamento. Além disso, como o controlador foi implementado utilizando como base a linguagem C/C++, essa aplicação pode ser transferida facilmente para qualquer controlador lógico programável (CLP) moderno, *hardware* mais adequado para aplicações industriais, já que ultimamente o suporte à linguagem C/C++ tem sido uma tendência nessas plataformas (Kufalor et al., 2014).

A interface entre o microcontrolador e o *software* MATLAB, no qual executou-se a integração do modelo não linear (1), foi estabelecida através da comunicação serial. Portanto, o fluxo de dados na aplicação é resumido como segue: as ações de controle do IHMPC são calculadas no microcontrolador e enviadas ao MATLAB para a simulação da planta virtual, após isso, os valores das saídas controladas são enviadas do MATLAB ao microcontrolador, a fim de realimentar a malha de controle. A Figura 2 descreve a aplicação proposta, onde as variáveis controladas são o *Head* da bomba e a pressão de *intake*, enquanto a frequência rotacional e a abertura da válvula *choke* são as variáveis manipuladas.

#### 3.2 Solver QP

Para solucionar o problema de otimização (7), resultante da lei de controle do IHMPC, um *solver* baseado no método dos conjuntos ativos foi também implementado no microcontrolador ESP32. Esse método é uma das abordagens mais populares para resolver problemas QP de pequena e média escala, e sua escolha se deu, sobretudo, pela possibilidade de obter-se uma solução de alta precisão dentro de poucas iterações e por não haver parâmetro de sintonia em sua implementação. Além disso, a quantidade de iterações desse método geralmente possui baixa variabilidade ao longo da execução do controlador, o que

não ocorre, por exemplo, com os chamados métodos de primeira ordem, nos quais durante os instantes de regime transitório a quantidade de iterações aumenta drasticamente (Peccin et al., 2020).

---

**Algoritmo 1:** *solver* baseado no método dos conjuntos ativos para a QP (7)

---

**Entrada:**  $x_0$  viável,  $H$ ,  $c$ ,  $A$ ,  $b$

**Saída:**  $x_k$

```

1  $k \leftarrow 0$ ;
2 defina um conjunto de trabalho inicial  $W_0$ ;
3 enquanto algum  $\lambda_i < 0, i \in W_k$ , ou  $d_k \neq 0$  faça
4     Atualize a matriz de restrições ativas:  $A_k$ , cujas linhas são  $A_i, i \in W_k$ ;
5     calcule o gradiente no ponto atual:  $g_k = Hx_k + c$ ;
6     resolva o sistema linear:
       
$$\begin{bmatrix} H & A_k^T \\ A_k & 0 \end{bmatrix} \cdot \begin{bmatrix} d_k \\ \lambda \end{bmatrix} = \begin{bmatrix} -g_k \\ 0 \end{bmatrix};$$

7     se as condições de KKT são satisfeitas então
8          $x_k \leftarrow x_k$ ;
9         pare (solução encontrada)
10    senão
11        calcule:
           
$$\alpha_k = \min_{i \notin W_k, A_i d_k > 0} \left( 1, \frac{b_i - A_i x_k}{A_i d_k} \right);$$

12         $x_k \leftarrow x_k + \alpha_k d_k$ ;
13        Se  $\alpha_k < 1$ , há alguma restrição ativa, adicione-a ao conjunto  $W_k$ ;
14    fim
15     $k \leftarrow k + 1$ ;
16 fim
```

---

O *solver* AS (*Active Set*) foi implementado baseado no Algoritmo 1, e como pode ser visto, é necessário que uma estimativa inicial viável ( $x_0$ ) seja fornecida. Basicamente, o algoritmo forma um subconjunto ( $A$ ) contendo apenas as restrições ativas do problema e usa o gradiente e as informações dos multiplicadores de Lagrange para remover ou adicionar uma nova restrição. Este subconjunto é modificado até que as condições de otimalidade sejam satisfeitas ou o limite máximo de iterações seja atingido. Uma importante propriedade desse algoritmo é que a cada iteração uma solução viável é obtida. Sendo assim, caso ocorra alguma parada antecipada, garante-se ao menos uma solução que atenda às restrições do problema de otimização.

Vale ressaltar que a funcionalidade do algoritmo 1 está condicionada à solução do sistema linear apresentado no passo 6. No entanto, durante a modificação da matriz de restrições ativas ( $A_k$ ), esta pode eventualmente tornar-se uma

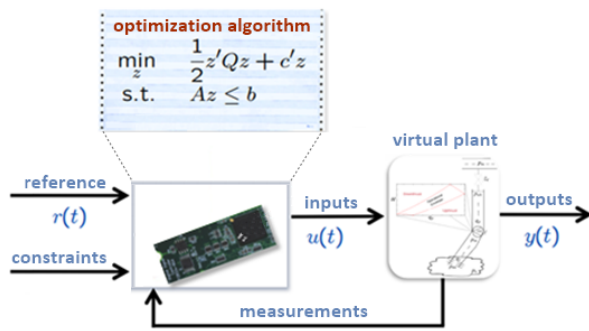


Figura 2: Esquema para a aplicação *hardware-in-the-loop*.

matriz singular, conseqüentemente, o sistema linear não poderá ser solucionado com os métodos numéricos convencionais. Para contornar essa limitação, utilizou-se um método de solução baseado na decomposição QR que viabiliza a solução do sistema linear mesmo quando a matriz  $A_k$  é singular. Os detalhes dessa abordagem não serão discutidos neste trabalho, mas podem ser vistos em (Das and Fuller, 2008).

#### 4 Resultados

Esta seção apresenta os resultados obtidos através do teste *hardware-in-the-loop* com o ESP32, avaliando um cenário típico de operação de um poço de petróleo operado por BCS. A fim de validar o algoritmo embarcado no ESP32, realizou-se uma comparação com os resultados de simulação obtidos via MATLAB, utilizando seu *solver* Quadprog (conjuntos ativos) para o cálculo da lei de controle e usando-o como *benchmark*. Os resultados de simulação do MATLAB foram obtidos em um computador com processador Intel Core i3 2.2GHz, 4 GB de memória RAM e sistema operacional Windows 10.

Os parâmetros de sintonia do IHMPC foram ajustados em:  $\Delta t = 1$  s,  $m = 3$ ,  $\mathbf{Q}_y = \text{diag}([1, 1])$ ,  $\mathbf{R} = \text{diag}([1, 1])$ ,  $\mathbf{Q}_u = \text{diag}([0, 1])$ ,  $\mathbf{S}_u = \text{diag}([0, 100])$ ,  $\mathbf{S}_y = \text{diag}([1, 1] \times 10^6)$ ,  $\mathbf{u}_{\min} = [35 \text{ Hz}, 0 \text{ \%}]$ ,  $\mathbf{u}_{\max} = [65 \text{ Hz}, 100 \text{ \%}]$ ,  $\Delta \mathbf{u}_{\max} = [0.5 \text{ Hz}, 1 \text{ \%}]$ . Com essa sintonia, recaiu-se em um problema de otimização com 8 variáveis de decisão ( $x \in \mathbb{R}^8$ ,  $H \in \mathbb{S}_+^{8 \times 8}$ ,  $c \in \mathbb{R}^8$ ) e 28 restrições de desigualdade ( $A \in \mathbb{R}^{28 \times 8}$ ,  $b \in \mathbb{R}^{28}$ ). Além disso, o modelo linearizado utilizado no IHMPC foi obtido no ponto de equilíbrio  $p_{in,ss} = 6.0 \times 10^6$  Pa,  $H_{ss} = 592.12$  m,  $f_{ss} = 50$  Hz,  $z_{c,ss} = 50\%$  e  $p_m,ss = 2.0 \times 10^6$  Pa, e discretizado para um período de amostragem igual a 1 s.

As dinâmicas resultantes no cenário de rastreamento de referência simulado, para a pressão de *intake* e para as entradas, são mostradas nas Figuras 3 e 4, respectivamente. Como pode ser

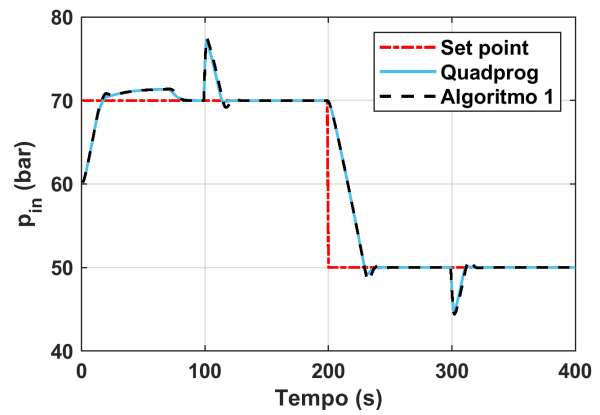


Figura 3: Dinâmica da pressão de *intake*.

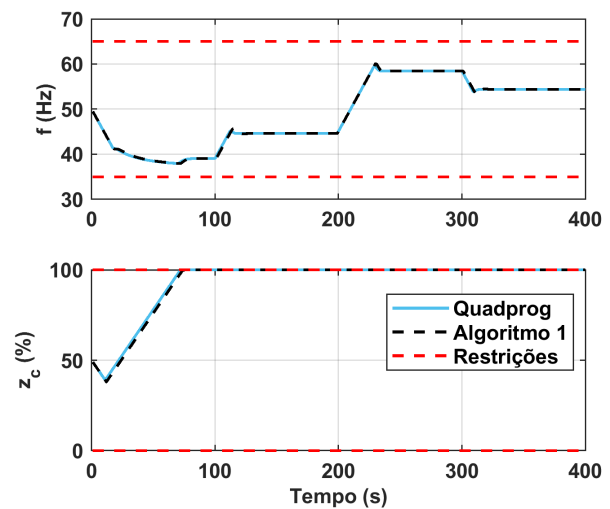


Figura 4: Sinal das variáveis manipuladas.

visto, o controlador é capaz de conduzir a pressão de *intake* para as referências desejadas com um bom desempenho de rastreamento para ambos os algoritmos testados, conduzindo também a abertura da *choke* de produção ( $z_c$ ) para seu alvo econômico, qual seja,  $\mathbf{u}_{tg} = 100\%$ . Note ainda que o controlador é suficientemente robusto para realizar a compensação de distúrbios que foram aplicados nos instantes 100 s e 300 s, e que foram produzidos através de variações na pressão de *manifold*.

Com relação ao envelope operacional, a Figura 5 mostra que o controlador mantém o sistema dentro da região segura de operação, condição desejada para preservar a vida útil dos equipamentos que compõem o sistema BCS. Note também que o ponto de operação ao fim da simulação aproxima-se da condição de *upthrust*, sem violá-la, onde geralmente estão localizados os pontos de maior rentabilidade econômica.

No que concerne ao algoritmo de conjuntos ativos implementado para solucionar a QP, po-

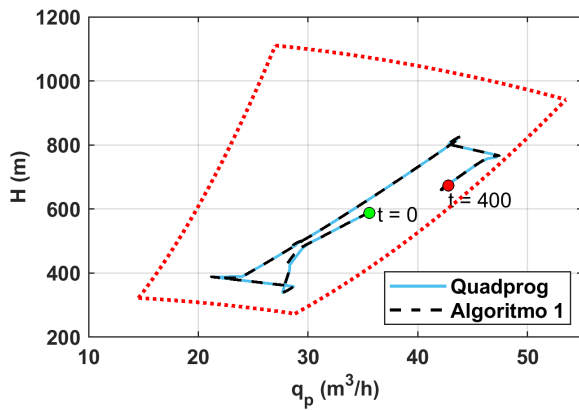


Figura 5: Envelope operacional do BCS.

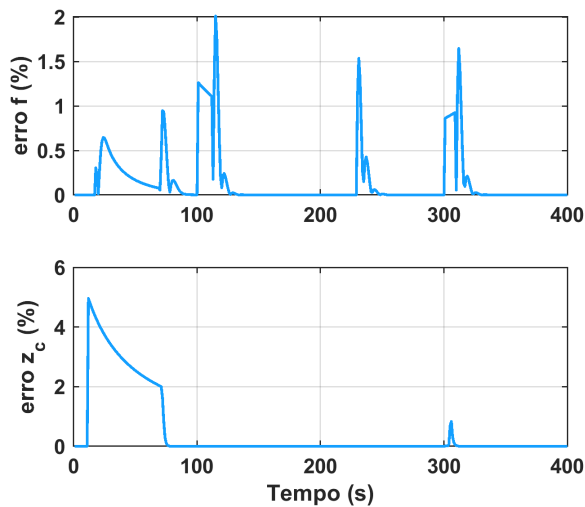


Figura 6: Erro relativo das trajetórias das variáveis manipuladas.

demos notar que as trajetórias das variáveis manipuladas obtidas com o Algoritmo 1 coincidem com aquelas obtidas a partir da Quadprog no MATLAB, sendo essas consideradas como *benchmark*. Os erros relativos das trajetórias das variáveis manipuladas são mostrados na Figura 6, indicando um baixo percentual de erro que ocorre somente durante os transientes, validando portanto o algoritmo de otimização embarcado no ESP32.

A comparação de desempenho dos algoritmos de otimização é ilustrada nas Figuras 7 e 8. É importante mencionar que a comparação entre os tempos de computação somente é usada como uma análise qualitativa do algoritmo embarcado e do desempenho de processamento do ESP32, já que os algoritmos foram executados em *hardwares* distintos. A Figura 7 mostra claramente que o tempo de computação para o Algoritmo 1 (ESP32) é menor do que para a Quadprog ao longo da execução do controlador. O tempo médio de computação do algoritmo embarcado foi 2.93 ms, enquanto a Quadprog levou em média 5.75 ms para executar o

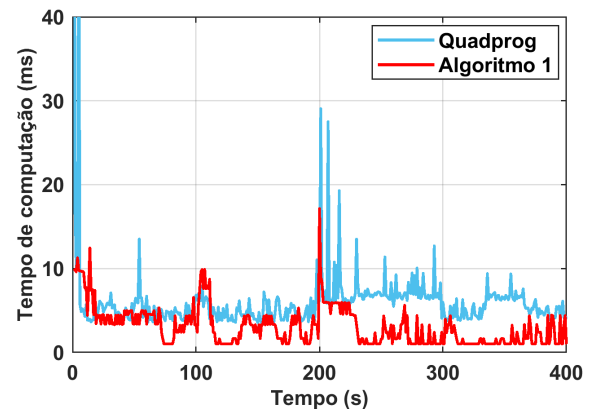


Figura 7: Comparação dos tempos de cômputo entre os algoritmos.

cálculo da lei de controle, ambas atendendo muito bem a taxa de amostragem exigida de 1 Hz. Logo no início da simulação, há um alto tempo de computação com a Quadprog devido ao carregamento de variáveis na memória, por isso, esse intervalo de tempo foi desconsiderado nos dados apresentados anteriormente. Com o Algoritmo 1, o pior caso de tempo de computação no cenário de controle simulado foi medido em 17.17 ms, que corresponde apenas a 1.72 % do tempo de amostragem da aplicação.

Finalmente, a Figura 8 mostra o número de iterações dos algoritmos, os quais foram limitados a executarem no máximo 50 iterações. Claramente o Algoritmo 1 possui um desempenho superior em relação à Quadprog, pois converge dentro de poucas iterações tanto em condições de estado estacionário, quanto em períodos transientes. Com isso, o tempo de cálculo do *solver* pode ser facilmente previsível, o que é crucial em aplicações de tempo real. Neste cenário de controle simulado, o Algoritmo 1 executou no máximo 16 iterações, registrada no instante 200 s durante a mudança de referência. Já para a Quadprog, registrou-se um máximo de 50 iterações, atingindo o limite máximo especificado para os algoritmos. Para ambos os algoritmos, a tolerância dos resíduos foi configurada em  $10^{-10}$ .

## 5 Conclusões

Este trabalho apresentou uma implementação embarcada de um controlador preditivo de horizonte infinito aplicado a um sistema BCS, avaliando os aspectos práticos a partir do teste *hardware-in-the-loop* com o microcontrolador ESP32. A estratégia de controle por faixa, proposta para atender aos objetivos de controle e econômico do sistema, mostrou-se capaz de fornecer uma solução sistemática e com bom desempenho no cenário típico de operação simulado, além de manter o sistema dentro do envelope operacional, condição funda-



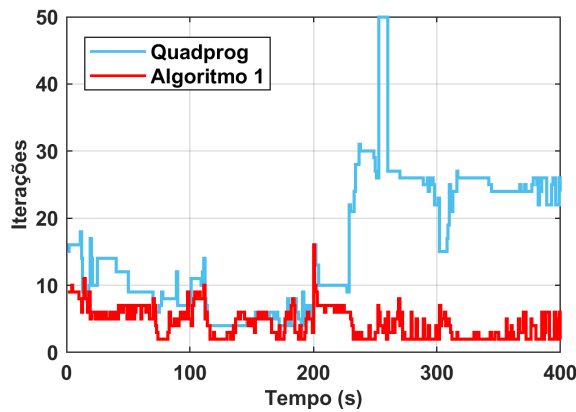


Figura 8: Comparação da quantidade de iterações entre os algoritmos.

mental para manter a vida útil dos equipamentos do BCS.

Os resultados mostraram que o algoritmo de otimização embarcado forneceu uma solução com boa precisão e convergiu dentro de poucas iterações, mesmo em condições de regime transiente, mostrando-se adequado para aplicações em tempo real. O algoritmo foi validado usando como *benchmark* resultados de simulação obtidos a partir da Quadprog, fornecendo desempenho superior do que o *solver* do MATLAB. Por fim, destaca-se a potencialidade de extensão desta implementação para um *hardware* industrial compatível com a linguagem C/C++, como os CLPs de última geração.

#### Agradecimentos

O presente trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) e do Programa de Recursos Humanos da Agência Nacional do Petróleo, Gás Natural e Biocombustíveis (ANP) - PRH-35.1.

#### Referências

Binder, B. J. T., Kufoalor, D. K. M. and Pavlov, J. (2014). Embedded Model Predictive Control for an Electric Submersible Pump on a Programmable Logic Controller, *IEEE Conference on Control Applications (CCA)*, number 2, pp. 579 – 585.

Binder, B., Johansen, T. and Imsland, L. (2019). Improved predictions from measured disturbances in linear model predictive control, *Journal of Process Control* **75**: 86–106.

Camacho, E. F. and Bordons, C. (2004). *Model Predictive Control*, 2 edn, Springer Verlag.

Das, I. and Fuller, J. W. (2008). Real-time quadratic programming for control of dynamical systems, *European Patent Office, EP1426840A1*.

Delou, P. A., Azevedo, J. P., Krishnamoorthy, D., De Souza, M. B. and Secchi, A. R. (2019). Model predictive control with adaptive strategy applied to an electric submersible pump in a subsea environment, *IFAC-PapersOnLine* **52**(1): 784–789.

Fontes, R. M., Costa, E. A., Abreu, O. S. L., Martins, M. A. F. and Schnitman, L. (2020). On application of a zone IHMPC to an ESP-lifted oil well system, *XXIII Brazilian Congress on Automation*.

González, A. H. and Odloak, D. (2009). A stable MPC with zone control, *Journal of Process Control* **19**(1): 110–122.

Krishnamoorthy, D., Bergheim, E. M., Pavlov, A., Fredriksen, M. and Fjalestad, K. (2016). Modelling and Robustness Analysis of Model Predictive Control for Electrical Submersible Pump Lifted Heavy Oil Wells, *IFAC-PapersOnLine* **49**(7): 544–549.

Kufoalor, D. K. M., Richter, S., Imsland, L., Johansen, T. A., Morari, M. and Eikrem, G. O. (2014). Embedded model predictive control on a plc using a primal-dual first-order method for a subsea separation process, *22nd Mediterranean Conference on Control and Automation, IEEE*, pp. 368–373.

Liang, X., He, J. and Du, L. (2015). Electrical Submersible Pump System Grounding: Current Practice and Future Trend, *IEEE Transactions on Industry Applications* **51**(6): 5030–5037.

Odloak, D. (2004). Extended robust model predictive control, *AIChE Journal* **50**(8): 1824–1836.

Pavlov, A., Krishnamoorthy, D., Fjalestad, K., Aske, E. and Fredriksen, M. (2014). Modelling and model predictive control of oil wells with Electric Submersible Pumps, *2014 IEEE Conference on Control Applications (CCA)*, number 3905, pp. 586–592.

Peccin, V., Martins Lima, D., Flesch, R. and Normey-Rico, J. (2020). Fast algorithms for constrained generalised predictive control with on-line optimisation, *IET Control Theory Applications* **15**.

Takacs, G. (2009). *Electrical Submersible Pumps Manual*, 1st edn, Gulf Professional Publishing.