DOI: 10.48011/asba.v2i1.1652

Algoritmos GPC de Cômputo Rápido com Métodos de Ponto Interior e Programação Quadrática Sem Projeção

Vinícius Berndsen Peccin * **Daniel Martins Lima** **
Rodolfo César Costa Flesch * **Julio Elias Normey-Rico** *

* *Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina, Florianópolis, Santa Catarina, Brasil (e-mail: vinicius.peccin@ifsc.edu.br).*
** *Departamento de Controle, Automação e Computação, Universidade Federal de Santa Catarina, Blumenau, Santa Catarina, Brasil (e-mail: daniel.lima@ufsc.br)*

Abstract: Historically, generalized predictive control (GPC) has been used in plants with slow dynamics, due to the high computational cost required by the control action code. More recently, fast GPC solutions have been presented. This work aims to contribute with two algorithms based on the interior point with barrier method (IP) and the method of parallel quadratic programming (PFQP). A comparative study with two GPC algorithms from the literature which make use of the accelerated dual gradient-projection method (GPAD) and alternating direction method of multipliers (ADMM) is also presented. The commercial optimizer named Gurobi was used as a reference. From the comparative study, it was possible to verify a better performance of the GPAD algorithm and the sensitivity of the ADMM and PFQP algorithms in scenarios with smaller tolerances and different types of constraints. It was also emphasized that, although slower, the algorithm with IP showed low variation in the computation time with the proposed scenarios.

Resumo: Historicamente, o controle preditivo generalizado (GPC) vem sendo mais utilizado em plantas com dinâmicas lentas, devido ao alto custo computacional requerido para o cômputo da ação de controle. Mais recentemente, soluções de cômputo rápido do sinal de controle do GPC vêm sendo apresentadas. Nesse sentido, o presente trabalho visa contribuir com a proposta de dois algoritmos baseados no método de ponto interior (IP) e no método de programação quadrática paralela (PFQP). É também apresentado um estudo comparativo com dois algoritmos de GPC da literatura que utilizam o método de gradiente projetado acelerado dual (GPAD) e o método de multiplicadores em direções alternadas (ADMM). Para servir de referência, foi utilizado o otimizador comercial Gurobi. A partir do estudo comparativo, pôde-se verificar um melhor desempenho do algoritmo GPAD e a sensibilidade dos algoritmos ADMM e PFQP em cenários com tolerâncias menores e diferentes tipos de restrição. Ressaltou-se também que, apesar de mais lento, o algoritmo com IP apresentou baixa variação no tempo de cômputo com os cenários propostos.

Keywords: ADMM; generalized predictive control; GPAD; interior point; PFQP.
Palavras-chaves: ADMM; controle preditivo generalizado; GPAD; PFQP; ponto interior.

1. INTRODUÇÃO

O controle preditivo baseado em modelo (MPC, do inglês *Model Predictive Control*) é uma das técnicas de controle avançado mais populares. Uma das principais vantagens dessa técnica é o tratamento explícito de restrições. Entretanto, devido ao custo computacional relativamente elevado para o cômputo do sinal de controle, o MPC é mais utilizado para plantas com dinâmicas mais lentas. Atualmente, diversas pesquisas abordam algoritmos para a aceleração do cômputo do MPC. Entretanto, a grande maioria dos trabalhos utiliza a formulação do MPC em espaço de estados (SSMPC, do inglês *State Space Model Predictive Control*), que é uma formulação popular na academia.

Mais recentemente, trabalhos como Peccin et al. (2019) e Peccin et al. (2020) apresentam uma formulação do controle preditivo generalizado (GPC, do inglês *Generalized Predictive Control*) com métodos de otimização *online* para aceleração do tempo de cômputo. O GPC é uma formulação MPC bastante popular na indústria.

⋆ Este trabalho foi apoiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), projetos 304032/2019-0, 309244/2018-8 e 421120/2016-9, e pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES/Brazil) no programa PrInt/UFSC.

Sua popularidade, em grande parte, se dá pois o modelo utilizado pelo GPC para a predição da saída baseia-se na função de transferência discreta, que por sua vez engloba conceitos bastante populares no meio industrial (Camacho and Bordons, 2004).

Em geral, pode-se dizer que uma das principais formas de aceleração do cômputo da ação de controle, no contexto de MPC, é a partir da aceleração da solução do problema de programação quadrática (QP, do inglês *Quadratic Programming*) que precisa ser resolvido a cada iteração. Uma abordagem, utilizada em SSMPC, se dá pela reformulação do QP em um problema de programação multiparamétrica e posterior cômputo da solução *offline* baseado em regiões. Esta abordagem é chamada de MPC explícito sua formulação é apresentada em Bemporad et al. (2002). Uma das limitações dessa abordagem é que o número de regiões cresce muito rapidamente com o aumento das dimensões do problema, inviabilizando a sua aplicação em problemas com características industriais típicas. Em Kvasnica et al. (2015), uma solução de SSMPC explícito sem regiões é apresentada de modo a mitigar essa limitação. Entre as soluções de cômputo *online*, no contexto de SSMPC, destacam-se os métodos de conjunto ativo, de ponto interior (IP, do inglês *Interior Point*), e os métodos de primeira ordem. Soluções SSMPC com conjunto ativo podem ser vistas em Cimini and Bemporad (2017) e Herceg et al. (2015). Entre os métodos de ponto interior, destaca-se o método de barreira (Roldao-Lopes et al., 2009 e Wills et al., 2011). Já entre os métodos de primeira ordem podem ser citados os trabalhos de Patrinos and Bemporad (2014), Boyd et al. (2011), O'Donoghue et al. (2013), Pu et al. (2017) e Cairano et al. (2013).

No contexto do GPC de cômputo rápido, em Peccin et al. (2019) foi apresentada uma formulação rápida baseada no método de gradiente projetado acelerado dual (GPAD, do inglês *Accelerated Dual Gradient-Projection*). Em Peccin et al. (2020) o algoritmo GPC foi apresentado com o método dos multiplicadores em direções alternadas (ADMM, do inglês *Alternating Direction Method of Multipliers*). Ambas as soluções são baseadas em algoritmos de primeira ordem. Em geral, os algoritmos de primeira ordem, apesar de rápidos e simples de computar, sofrem com a baixa precisão das soluções apresentadas e com a dificuldade de tratamento de algumas classes de restrições (Ferreau et al., 2017). Portanto, um dos objetivos deste trabalho é apresentar outras opções de algoritmos para GPC de modo a se obter soluções com características distintas e um estudo de comparação desses algoritmos com o GPAD e o ADMM para um problema específico. Um dos algoritmos apresentados baseia-se no método IP de barreira, que é bastante conhecido por resolver um QP em poucas iterações com boa precisão, entretanto ele requer operações algébricas mais complexas, que podem levar a maiores tempos de cômputo por iteração. O outro algoritmo baseia-se no método de programação quadrática sem projeção (PFQP, do inglês *Projection-Free Quadratic Programming*), que não necessita do passo de projeção da solução para o domínio das restrições, como no caso do GPAD (Cairano et al., 2013). Dessa forma, o PFQP pode apresentar um desempenho superior para determinados tipos de restrições. Assim, as principais contribuições deste artigo são:

- a apresentação de dois algoritmos GPC de cômputo rápido baseados no método de barreira e no método PFQP;
- a comparação de desempenho dos métodos propostos com os algoritmos GPCADMM e GPCGPAD por meio de simulação.

A continuação do artigo está organizada da seguinte forma: na Seção 2 é apresentada uma breve descrição da formulação GPC. Na Seção 3 os algoritmos do GPC com os métodos de barreira e PFQP são apresentados. Na Seção 4 é feita uma comparação dos algoritmos a partir de um estudo de caso simulado em MATLAB. A Seção 5 conclui o artigo.

2. CONTROLE PREDITIVO GENERALIZADO

O método chamado controle preditivo generalizado foi proposto por Clarke et al. (1987) e se tornou um dos métodos mais populares tanto na indústria quanto na academia (Camacho and Bordons, 2004). Para obter a predição da saída, o GPC utiliza o modelo autorregressivo com média móvel, integrador e entrada controlada (CARIMA, do inglês *Controlled Autoregressive Integrated Moving Average*).

A função custo típica do GPC, para o caso SISO, é dada por:

$$J_{GPC} = \sum_{j=N_1}^{N_2} \delta(j) \left[\hat{y}(k+j|k) - w(k+j)\right]^2 + \sum_{j=1}^{N_u} \lambda(j) \left[\Delta u(k+j-1)\right]^2, \tag{1}$$

onde $\hat{y}(k+j|k)$ representa uma predição da saída em um tempo futuro $k+j$, dado em k, $w(k+j)$ a referência em $k+j$, $\Delta u(k+j-1)$ o incremento de controle em $k+j-1$, $\delta(j)$ e $\lambda(j)$ ponderam o comportamento futuro do erro e do esforço de controle respectivamente, N_1 e N_2 definem o horizonte de predição e N_u é o horizonte de controle.

Para o cômputo da ação de controle, no caso com restrições, o problema de programação quadrática típico de um GPC, como apresentado em Camacho and Bordons (2004), pode ser escrito como:

$$\begin{aligned} \min_{\Delta \mathbf{u}} \quad & \frac{1}{2}\Delta \mathbf{u}^T \mathbf{H} \Delta \mathbf{u} + \mathbf{b}^T \Delta \mathbf{u} \\ \text{s.a.} \quad & \bar{\mathbf{R}} \Delta \mathbf{u} \leq \bar{\mathbf{r}}, \end{aligned} \tag{2}$$

onde $\bar{\mathbf{R}} \in \mathbb{R}^{n_{rin} \times N_u}$ representa a matriz de restrições, $\bar{\mathbf{r}} \in \mathbb{R}^{n_{rin}}$ o vetor de restrições e n_{rin} o número de restrições de desigualdade.

A obtenção das matrizes $\mathbf{H}$, $\mathbf{b}$, $\bar{\mathbf{R}}$, $\bar{\mathbf{r}}$ é detalhada tanto em Camacho and Bordons (2004) quanto em Normey-Rico and Camacho (2007). O algoritmo rápido do GPC que foi utilizado neste trabalho é apresentado em Peccin et al. (2019) e Peccin et al. (2020).

3. PROPOSTAS DE TÉCNICAS DE OTIMIZAÇÃO

A partir do QP do GPC, apresentado em (2), os algoritmos com os métodos de ponto interior com barreira e programação quadrática sem projeção são apresentados.

3.1 Método de ponto interior com barreira (IP)

O método IP tem como objetivo aproximar um problema de otimização com restrições de desigualdade a um problema com apenas restrições de igualdade. Isso é feito por meio da inclusão das restrições de desigualdade na função objetivo, de modo que ela assuma valores muito altos nos casos em que uma restrição é violada. Para incluir as restrições de desigualdade na função objetivo de um problema como (2), pode ser utilizada um função indicadora logarítmica do tipo:

$$\hat{I}_- = -(1/t)\log(-\bar{x}), \quad \mathbf{dom}\ \hat{I}_- = \mathbb{R}^-, \tag{3}$$

onde $t > 0$ é um parâmetro que ajusta a exatidão da aproximação. Aplicando $\hat{I}_-$ em (2) obtém-se o problema convexo irrestrito:

$$\begin{aligned} \min_{\mathbf{\Delta u}} \quad & \frac{1}{2}\mathbf{\Delta u}^T\mathbf{H}\mathbf{\Delta u} + \mathbf{b}^T\mathbf{\Delta u} + \\ & + \sum_{i=0}^{n_{rin}} -(1/t)\log(-\bar{\mathbf{R}}_{li}\mathbf{\Delta u} + \bar{\mathbf{r}}_i), \end{aligned} \tag{4}$$

onde $\bar{\mathbf{R}}_{li}$ representa a i-ésima linha da matriz $\bar{\mathbf{R}}$ e $\bar{\mathbf{r}}_i$ o i-ésimo elemento do vetor $\bar{\mathbf{r}}$. Dessa forma, o método de barreira se baseia na resolução de uma sequência de problemas de minimização irrestritos, utilizando o último ponto encontrado como o ponto de início da próxima iteração. A cada iteração o valor de t é incrementado, até que $t \geq n_{rin}/\epsilon$, o que garante que seja obtida uma solução ϵ-subótima do problema original (Boyd and Vandenberghe, 2004).

Para resolver (4), apesar de qualquer método de minimização linear com restrições de igualdade poder ser utilizado, em geral, utiliza-se o método de Newton que garante uma solução factível a cada iteração (Boyd and Vandenberghe, 2004). Esse cômputo é chamado de passo de centralização, uma vez que o caminho central é definido como o conjunto de pontos chamados pontos centrais. Os pontos centrais são caracterizados por atenderem as condições necessárias e suficientes para serem considerados pontos estritamente factíveis. Dessa forma, o passo de Newton é dado por:

$$\Delta\mathbf{x}_{nt} = -\nabla^2 f(\mathbf{x})^{-1}\nabla f(\mathbf{x}).$$

O gradiente $\mathcal{G}$ e a Hessiana $\mathcal{H}$ da função custo de (4) são dadas por:

$$\mathcal{G} = \mathbf{H}\mathbf{\Delta u} + \mathbf{b} + \sum_{i=0}^{n_{rin}} \frac{(1/t)}{\bar{\mathbf{R}}_{li}\mathbf{\Delta u} - \bar{\mathbf{r}}_i}\bar{\mathbf{R}}_{li}^T$$

$$\mathcal{H} = \mathbf{H} + \sum_{i=0}^{n_{rin}} \frac{(1/t)}{(\bar{\mathbf{R}}_{li}\mathbf{\Delta u} - \bar{\mathbf{r}}_i)^2}\bar{\mathbf{R}}_{li}^T\bar{\mathbf{R}}_{li}.$$

Portanto, o passo de Newton para o problema (4) é:

$$\Delta\mathbf{x}_{nt} = -\mathcal{H}^{-1}\mathcal{G}.$$

O algoritmo de Newton aplicado ao problema (4) pode ser visto no Algoritmo 1.

Ao se avaliar o Algoritmo 1, pode-se constatar que o cômputo mais custoso se refere ao passo de Newton. Pode-se abstrair esse passo como o cômputo de um sistema linear do tipo $\mathbf{Ax} = \mathbf{b}$, que tipicamente requer $O(N^3 n_{rin}^3)$ operações em um processador serial (Wills et al., 2011). Esse tipo de sistema poderia ser resolvido de forma eficiente por meio

Algoritmo 1: Método de Newton

Entrada: $\Delta\mathbf{u}^0$ estritamente factível, t, $\mathbf{H}$, $\mathbf{b}$, $\bar{\mathbf{R}}$, $\bar{\mathbf{r}}$
Saída: $\Delta\mathbf{u}$
Dados: L_{nt}
início
  para $i = 1 : L_{nt}$ **faça**
    Atualiza $\mathcal{G} := \mathbf{H}\mathbf{\Delta u} + \mathbf{b} + \sum_{i=0}^{n_{rin}} \frac{(1/t)}{\bar{\mathbf{R}}_{li}\mathbf{\Delta u} - \bar{\mathbf{r}}_i}\bar{\mathbf{R}}_{li}^T$;
    Atualiza $\mathcal{H} := \mathbf{H} + \sum_{i=0}^{n_{rin}} \frac{(1/t)}{(\bar{\mathbf{R}}_{li}\mathbf{\Delta u} - \bar{\mathbf{r}}_i)^2}\bar{\mathbf{R}}_{li}^T\bar{\mathbf{R}}_{li}$;
    Computa o passo de Newton: $\Delta\mathbf{x}_{nt} = -\mathcal{H}^{-1}\mathcal{G}$;
    Escolhe o tamanho do passo l por busca retroativa linear ;
    Atualiza $\Delta\mathbf{u} = \Delta\mathbf{u} + l\Delta\mathbf{x}_{nt}$;
fim

da decomposição das matrizes e exploração da estrutura da matriz $\mathbf{A}$, como a decomposição de Cholesky. Entretanto, a decomposição só é eficiente se a estrutura de $\mathbf{A}$ for esparsa. Para o caso do GPC, a matriz tipicamente é densa, então não é trivial explorar a estrutura. Dessa forma, optou-se por um método iterativo. O método escolhido foi o do gradiente conjugado (Golub and Van Loan, 1996). Essa abordagem foi utilizada no contexto de SSMPC e FPGA em Roldao-Lopes et al. (2009) e Wills et al. (2011). Para entender como o método do gradiente conjugado pode ser utilizado para resolver o problema $\mathbf{Ax} = \mathbf{b}$, pode-se considerar o problema de minimização:

$$\min_{\mathbf{x}} \quad f_{gc}(x) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{x}^T\mathbf{b}, \tag{5}$$

com $\mathbf{A} \in \mathbb{R}^{n\times n}$ simétrica positiva definida e $\mathbf{b} \in \mathbb{R}^n$. O mínimo de f_{gc} é $-\mathbf{b}^T\mathbf{A}^{-1}\mathbf{b}/2$, com $x^* = \mathbf{A}^{-1}\mathbf{b}$. Dessa forma, minimizar f_{gc} e resolver $\mathbf{Ax} = \mathbf{b}$ são problemas equivalentes. O algoritmo 2 apresenta o método do gradiente conjugado com busca exata de linha (em inglês, *exact line search*) aplicado ao cômputo do passo de Newton.

Algoritmo 2: Método do Gradiente Conjugado

Entrada: $\Delta\mathbf{x}_{nt}^0$ estritamente factível, $\mathcal{H}$, $\mathcal{G}$
Saída: $\Delta\mathbf{x}_{nt}$
início
  Calcula o resíduo inicial: $\mathbf{r} = \mathcal{G} - \mathcal{H}\Delta\mathbf{x}_{nt}^0$;
  i = 0;
  enquanto $\mathbf{r} \neq 0$ **faça**
    $i = i + 1$;
    se $i = 1$ **então**
      Calcula a direção do passo: $\mathbf{p} = \mathbf{r}$;
    senão
      Calcula a direção do passo: $\mathbf{p} = \mathbf{r} + \frac{\mathbf{r}^T\mathbf{r}}{\mathbf{r}^{-T}\mathbf{r}^-}\mathbf{p}$;
    Calcula o tamanho do passo $\alpha = \mathbf{r}^T\mathbf{r}/\left(\mathbf{p}^T\mathcal{H}\mathbf{p}\right)$;
    Atualiza $\Delta\mathbf{x}_{nt} = \Delta\mathbf{x}_{nt} + \alpha\mathbf{p}$;
    Guarda valor anterior: $\mathbf{r}^- = \mathbf{r}$;
    Atualiza o resíduo: $\mathbf{r} := \mathbf{r} - \alpha\mathcal{H}\mathbf{p}$;
fim

A partir dos métodos apresentados, pôde-se chegar ao algoritmo completo do método de barreira proposto para o GPC (GPCIP), que pode ser visto no Algoritmo 3.

Algoritmo 3: Controle preditivo generalizado (GPC) com método de barreira

Entrada: A, B, d, $\bar{\mathbf{R}}$, $\bar{\mathbf{r}}$

Saída: $u(k)$

Dados: λ, δ, N_u, N_1, N_2, $t := t_0 > 0$, $\mu > 1$, $\epsilon > 0$, $\beta \in (0,\ 1)$, $\varpi \in (0,\ 0,5)$, L_{gc}, L_{nt}.

início

$\tilde{A}(z^{-1}) = (1 - z^{-1})A(z^{-1})$;
para $i = 0 : N_2$ **faça**
  $g_i := z(1 - A(z^{-1}))g_{i-1} + B(z^{-1})\bar{u}_i$;
para $i = 1 : N_u$ **faça**
  $\mathbf{G}_{i:N,i} = \mathbf{g}_{N_1:N_2+1-i}$;
$\mathbf{H} = 2(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})$;
$k = 0$;
enquanto *modo automático* **faça**
  se *tempo de amostragem* **então**
    Adquire saída $y(k)$ e referência $w(k)$;
    $f_0 = y(k)$;
    para $i = 0 : N_2$ **faça**
      $f_{i+1} := z(1 - \tilde{A}(z^{-1}))f_i + B(z^{-1})\Delta u(k - d + i)$;
    $\mathbf{b}^T = 2(\mathbf{f}_{N_1:N_2} - \mathbf{w}_{N_1:N_2})^T\mathbf{G}$;
    enquanto $n_{rin}/t > \epsilon$ **faça**
      para cada $i = 1 : L_{nt}$ **faça**
        $\mathcal{G} := \mathbf{H}\Delta\mathbf{u} + \mathbf{b} + \sum_{i=0}^{n_{rin}} \frac{(1/t)}{\bar{\mathbf{R}}_{li}\Delta\mathbf{u} - \bar{\mathbf{r}}_i}\bar{\mathbf{R}}_{li}^T$;
        $\mathcal{H} := \mathbf{H} + \sum_{i=0}^{n_{rin}} \frac{(1/t)}{(\bar{\mathbf{R}}_{li}\Delta\mathbf{u} - \bar{\mathbf{r}}_i)^2}\bar{\mathbf{R}}_{li}^T\bar{\mathbf{R}}_{li}$;
        $\mathbf{r} = -\mathcal{G} - \mathcal{H}\Delta\mathbf{x}_{nt}$;
        para cada $j = 1 : L_{gc}$ **faça**
          se $j = 1$ **então**
            $\mathbf{p} = \mathbf{r}$;
          senão
            $\mathbf{p} = \mathbf{r} + \frac{\mathbf{r}^T\mathbf{r}}{\mathbf{r}^{-T}\mathbf{r}^-}\mathbf{p}$;
          $\alpha = \mathbf{r}^T\mathbf{r}/\mathbf{p}^T\mathcal{H}\mathbf{p}$;
          $\Delta\mathbf{x}_{nt} = \Delta\mathbf{x}_{nt} + \alpha\mathbf{p}$;
          $\mathbf{r}^- = \mathbf{r}$;
          $\mathbf{r} := \mathbf{r} - \alpha\mathcal{H}\mathbf{p}$;
        enquanto $\bar{\mathbf{R}}(\Delta\mathbf{u} + l\Delta\mathbf{x}_{nt}) - \bar{\mathbf{r}} > 0$ **faça**
          $l = \beta l$;
        enquanto
        $f(\Delta\mathbf{u} + l\Delta\mathbf{x}_{nt}) > f(\Delta\mathbf{u}) + l\varpi\mathcal{G}^T\Delta\mathbf{x}_{nt}$ **faça**
          $l = \beta l$;
        $\Delta\mathbf{u} = \Delta\mathbf{u} + l\Delta\mathbf{x}_{nt}$;
      $t := \mu t$;
    $\Delta u(k) = [1 \quad 0 \quad \cdots \quad 0]_{1\times N_u}\Delta\mathbf{u}$;
    $u(k) = u(k-1) + \Delta u(k)$;
    $k = k + 1$;

fim

Para a escolha da taxa de atualização de t, denotada por μ, é necessário ponderar entre o número de iterações na centralização, chamadas de iterações internas, ou de iterações no método propriamente dito, chamadas de iterações externas. Se μ for escolhido pequeno (próximo de 1), então a cada iteração externa, t é incrementado por um fator pequeno. Como resultado, o ponto gerado é um bom ponto inicial para o método de Newton, implicando um número reduzidos de passos de Newton. Por outro lado, o número de iterações externas é grande, uma vez que o resíduo é também reduzido por valores pequenos. Segundo Boyd and Vandenberghe (2004), na prática, para uma grande faixa de valores entre 3 e 100, aproximadamente, os dois efeitos praticamente se cancelam e o número de passos de Newton fica aproximadamente constante. Valores usuais de μ estão na faixa entre 10 e 20. A escolha de t_0 grande implica uma primeira iteração externa com um grande número de iterações internas. Já para a escolha de um t_0 pequeno, o algoritmo irá requerer uma quantidade extra de iterações externas. Portanto, já que n_{rin}/t_0 é o resíduo dual que resultará do primeiro passo de centralização, uma escolha razoável de t_0 é tal que n_{rin}/t_0 seja aproximadamente da mesma ordem de grandeza da tolerância de resíduo dual desejada ou μ vezes essa quantidade. O método de Newton, usado para computar o passo de centralização, foi limitado em L_{nt} iterações. Para o cômputo do passo de Newton foi utilizado o método do gradiente conjugado, limitado em L_{gc} iterações devido à rápida convergência. O tamanho do passo de Newton foi computado utilizando-se o algoritmo de busca retroativa linear, em duas partes. Na primeira, pode-se perceber uma busca por um ponto factível a cada iteração. Na segunda parte, a busca pela minimização da função custo, com o parâmetro ϖ sendo a fração aceitável de decaimento da função custo predita por extrapolamento linear. Em Boyd and Vandenberghe (2004) afirma-se que os valores típicos de ϖ são escolhidos entre $0,01$ e $0,3$, significando que é aceitável um decaimento da função custo de 1% a 30% da predição baseada na extrapolação linear.

3.2 Método programação quadrática sem projeção (PFQP)

O algoritmo proposto é baseado na aplicação do PFQP para a formulação paramétrica do MPC em espaço de estados apresentada em Cairano et al. (2013). Como o PFQP não necessita do passo de projeção para o domínio das restrições, o método se torna bastante competitivo em casos nos quais a projeção das restrições é complexa, como restrições de saída ou politópicas. A ideia principal do PFQP é, a cada passo do método, aproximar-se do valor ótimo a partir da direção do anti-gradiente e com o tamanho do passo escolhido para garantir que se mantenha na região factível.

Inicialmente, para adaptar o problema (2) para o formato padrão de mínimos quadrados não negativos, pode ser utilizada a formulação dual de (2):

$$\begin{aligned} \min_{\psi} \quad & \frac{1}{2}\psi^T\hat{\mathbf{H}}\psi + \hat{\mathbf{b}}^T\psi \\ \text{s.a.} \quad & \psi \geq \mathbf{0}, \end{aligned} \tag{6}$$

com $\hat{\mathbf{H}} = \bar{\mathbf{R}}\mathbf{H}^{-1}\bar{\mathbf{R}}^T$ e $\hat{\mathbf{b}} = \bar{\mathbf{R}}\mathbf{H}^{-1}\mathbf{b} + \bar{\mathbf{r}}$.

De modo a computar a solução ótima de (6), o Lagrangiano associado é dado por:

$$\mathcal{L}(\psi, \nu) = \frac{1}{2}\psi^T\hat{\mathbf{H}}\psi + \hat{\mathbf{b}}^T\psi - \nu^T\psi, \tag{7}$$

com $\nu \in \mathbb{R}^{n_{rin}}$ sendo o vetor de multiplicadores de Lagrange associados às restrições de desigualdade.

A partir de (7), a condição de otimalidade de (6), obtida por cálculo variacional, é dada por:

$$\psi \circ \nabla_{\psi}\mathcal{L}(\psi, \nu) = \mathbf{0}, \tag{8}$$

onde $\circ$ representa o produto de Hadamard entre dois vetores (multiplicação entre elementos na mesma posição).

Aplicando a condição de otimalidade (8) em (7) obtém-se:

$$\psi \circ (\hat{\mathbf{H}}\psi + \hat{\mathbf{b}} - \nu) = \mathbf{0}. \tag{9}$$

Se for definido $\hat{\mathbf{H}} = \hat{\mathbf{H}}^+ - \hat{\mathbf{H}}^-$ com $\hat{\mathbf{H}}^+ \triangleq \max(\mathbf{0}, \hat{\mathbf{H}})$, $\hat{\mathbf{H}}^- \triangleq \max(\mathbf{0}, -\hat{\mathbf{H}})$ e for aplicada a mesma lógica para $\hat{\mathbf{b}}$,

obtém-se:

$$\psi \circ ((\hat{\mathbf{H}}^{+}\psi + \hat{\mathbf{b}}^{+}) - (\hat{\mathbf{H}}^{-}\psi + \hat{\mathbf{b}}^{-} + \nu)) = \mathbf{0}, \tag{10}$$

que pode ser apresentado de forma reduzida como:

$$[\psi_{j+1}]_i = \frac{\left[\hat{\mathbf{H}}^{-}\psi_j + \hat{\mathbf{b}}^{-}\right]_i}{\left[\hat{\mathbf{H}}^{+}\psi_j + \hat{\mathbf{b}}^{+}\right]_i} [\psi_j]_i \,. \tag{11}$$

Para garantir a convergência, foi inserido um termo $\phi_{PFQP} \in \mathbb{R}^{n\times n}$

$$[\psi_{j+1}]_i = \frac{\left[(\hat{\mathbf{H}}^{-} + \phi_{PFQP})\psi_j + \hat{\mathbf{b}}^{-}\right]_i}{\left[(\hat{\mathbf{H}}^{+} + \phi_{PFQP})\psi_j + \hat{\mathbf{b}}^{+}\right]_i} [\psi_j]_i \,. \tag{12}$$

Em Cairano et al. (2013) foi provada a garantia de convergência com a escolha de ϕ_{PFQP} como uma matriz diagonal não negativa tal que os elementos da diagonal são dados por $[\phi_{PFQP}]_{ii} \geq \left[\hat{\mathbf{H}}^{-}\mathbf{1}\right]_i$, $\forall i = 1, .., n$.

A etapa de aceleração do PFQP, aplicada ao problema (6), é dada por:

$$l = \begin{cases} -\dfrac{(\hat{\mathbf{H}}\psi + \hat{\mathbf{b}})^T\mathbf{p}}{\mathbf{p}^T\hat{\mathbf{H}}\mathbf{p}} & \text{se } \mathbf{p}^T\hat{\mathbf{H}}\mathbf{p} > 0 \\ 0 & \text{senão} \end{cases}, \tag{13}$$

e $\mathbf{p}$, que representa a direção de descenso (valores negativos do gradiente), é definido como:

$$\mathbf{p} = \max(\mathbf{0}, -(\hat{\mathbf{H}}\psi + \hat{\mathbf{b}})). \tag{14}$$

O passo de aceleração deve ser executado a cada η_{PFQP} iterações do passo principal.

Para o critério de parada, podem ser utilizadas a factibilidade e a otimalidade primal. No caso do PFQP, a factibilidade primal é dada em termos duais, portanto:

$$-(\hat{\mathbf{b}} + \hat{\mathbf{H}}\psi) < \epsilon_{pri}. \tag{15}$$

Já para o critério de otimalidade primal, utilizando-se a propriedade da dualidade forte, aplicando a (2) e representando em termos duais, tem-se:

$$\psi^T\hat{\mathbf{H}}\psi + \hat{\mathbf{b}}^T\psi \leq \epsilon_{dual}. \tag{16}$$

A partir dos desenvolvimentos apresentados pode-se resumir o GPC com o PFQP (GPCPFQP) no Algoritmo 4.

Algoritmo 4: GPC com PFQP

Entrada: A, B, d, $\bar{\mathbf{R}}$, $\bar{\mathbf{r}}$
Saída: $u(k)$
Dados: λ, δ, N_u, N_1, N_2, η_{PFQP}, ϵ_{dual}, ϵ_{pri}
início
  $\tilde{A}(z^{-1}) = (1 - z^{-1})A(z^{-1})$;
  para $i = 0 : N_2$ **faça**
    $g_i := z(1 - A(z^{-1}))g_{i-1} + B(z^{-1})\bar{u}_i$;
  para $i = 1 : N_u$ **faça**
    $\mathbf{G}_{i:N,i} = \mathbf{g}_{N_1:N_2+1-i}$;
  $\mathbf{H} = 2(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})$;
  $\hat{\mathbf{H}} := \bar{\mathbf{R}}\mathbf{H}^{-1}\bar{\mathbf{R}}^T$;
  $\hat{\mathbf{H}}^{+} := \max(\mathbf{0}, \hat{\mathbf{H}})$; $\hat{\mathbf{H}}^{-} := \max(\mathbf{0}, -\hat{\mathbf{H}})$;
  $k = 0$;
  enquanto *modo automatico* **faça**
    se *tempo de amostragem* **então**
      Adquire saída $y(k)$ e referência $w(k)$;
      $f_0 = y(k)$;
      para $i = 0 : N_2$ **faça**
        $f_{i+1} := z(1 - \tilde{A}(z^{-1}))f_i + B(z^{-1})\Delta u(k - d + i)$;
      $\mathbf{b}^T = 2(\mathbf{f}_{N_1:N_2} - \mathbf{w}_{N_1:N_2})^T\mathbf{G}$;
      $\hat{\mathbf{b}} := \mathbf{b} + \bar{\mathbf{R}}\mathbf{H}^{-1}\mathbf{b}$;
      $\hat{\mathbf{b}}^{+} := \max(\mathbf{0}, \hat{\mathbf{b}})$; $\hat{\mathbf{b}}^{-} := \max(\mathbf{0}, -\hat{\mathbf{b}})$;
      $j := 0$;
      enquanto $\psi^T\hat{\mathbf{H}}\psi + \hat{\mathbf{b}}^T\psi \geq \epsilon_{dual}$ **ou** $-(\hat{\mathbf{b}} + \hat{\mathbf{H}}\psi) \geq \epsilon_{pri}$ **faça**
        se $j = \text{múltiplo}(\eta_{PFQP})$ **então**
          $p := \max(\mathbf{0}, -(\hat{\mathbf{H}}\psi + \hat{\mathbf{b}}))$;
          se $\mathbf{p}^T\hat{\mathbf{H}}\mathbf{p} > 0$ **então**
            $l := -\dfrac{(\hat{\mathbf{H}}\psi + \hat{\mathbf{b}})^T\mathbf{p}}{\mathbf{p}^T\hat{\mathbf{H}}\mathbf{p}}$
          senão
            $l := 0$;
          $\psi_{j+1} := \psi_j + l\mathbf{p}$;
        senão
          para $i = 1 : n_{rin}$ **faça**
            $[\psi_{j+1}]_i := \dfrac{\left[(\hat{\mathbf{H}}^{-} + \phi_{PFQP})\psi_j + \hat{\mathbf{b}}^{-}\right]_i}{\left[(\hat{\mathbf{H}}^{+} + \phi_{PFQP})\psi_j + \hat{\mathbf{b}}^{+}\right]_i} [\psi_j]_i$;
        $j := j + 1$;
      $\Delta\mathbf{u} := \mathbf{H}^{-1}(\mathbf{b} + \bar{\mathbf{R}}^T\psi)$;
      $\Delta u(k) = [1 \quad 0 \quad \cdots \quad 0]_{1\times N_u}\Delta\mathbf{u}$;
      $u(k) = u(k-1) + \Delta u(k)$;
      $k = k + 1$;
fim

4. COMPARAÇÃO DAS TÉCNICAS

Nesta seção é apresentada uma comparação dos algoritmos propostos com emprego de simulações em MATLAB®. Os algoritmos foram aplicados no mesmo estudo de caso apresentados em Peccin et al. (2019) e Peccin et al. (2020), cujo processo SISO é modelado pela função de transferência em tempo discreto:

$$G(z) = \frac{0,035z + 0,0307}{z^2 - 1,6375z + 0,6703}.$$

Um *solver* comercial chamado Gurobi foi utilizado como base de comparação para os algoritmos. O método escolhido para a solução do QP no solver Gurobi foi o de ponto interior com barreira (Gurobi, 2020). A simulação foi realizada por 150 amostras discretas e com três trocas de referência do tipo degrau. Os parâmetros de controle foram escolhidos com horizonte de controle $N_u = 5$, horizontes de predição $N_1 = 1$ e $N_2 = 20$, ponderação no esforço de controle $\lambda = 1$, ponderação no erro $\delta = 1$ e referências futuras $w(t + j)$ conhecidas. Não foi utilizado *warm-start* em nenhum dos algoritmos para se avaliar o pior caso de tempo de execução. O algoritmo GPCIP foi sintonizado com $t_0 = 0,1$, $\mu = 8$, $\beta = 0,4$, $\varpi = 0,02$, $L_{nt} = 3$ e $L_{gc} = 5$. O algoritmo GPCPFQP foi sintonizado com $\eta_{PFQP} = 5$ e o algoritmo GPCADMM com $\rho = 50$. Os tempos de cômputo durante a simulação foram obtidos a partir do comando *tic-toc* do MATLAB. O computador utilizado possui um processador Core i3 de 2,40 GHz e uma memória RAM de 12 GB.

O ensaio foi proposto com três cenários de modo a avaliar o comportamento dos algoritmos em relação ao tempo de cômputo e o número de iterações.

No cenário 1 foi inserida uma restrição no incremento de controle

$$|\Delta \mathbf{u}(k+j)| \leq 0,05,$$

com $j = 0, ..., N_u - 1$. Dessa forma, o problema de otimização recaiu então em uma matriz $\mathbf{H} \in \mathbb{R}^{5\times5}$ com 10 restrições. Todos os algoritmos foram sintonizados para uma tolerância dos resíduos de $\pm10^{-3}$.

No cenário 2 foi utilizada apenas uma restrição na variável de saída

$$0 \leq \bar{\mathbf{y}}(k+j) \leq 2,1$$

com $j = 1, ..., N-1$ e totalizando 40 restrições.

No cenário 3, foram utilizadas ambas as restrições, na variável de controle e na de saída, totalizando 50 restrições. Nesse cenário, as tolerâncias também foram alteradas para $\pm10^{-6}$.

Os resultados para o cenário 1 podem ser vistos na Figura 1. Os sinais de saída da planta e os incrementos de controle aplicados são apresentados na Figura 1a e verifica-se uma saturação atuando no incremento de controle em $\pm0,05$, evidenciando assim que as restrições estão ativas. Percebe-se na Figura 1b que os tempos de cômputo são menores para ADMM e o GPAD em relação ao PFQP e o IP. A partir do gráfico de iterações, percebe-se uma grande sensibilidade dos algoritmos de primeira ordem quando as restrições estão ativas, refletindo esse comportamento no aumento do número de iterações. Essa característica não aparece nos algoritmos IP e Gurobi. O IP, como esperado, teve um valor fixo de iterações do método de barreira, sem considerar as iterações internas do método de Newton e do gradiente dual. Este cenário é propício para os métodos de primeira ordem que apresentam um bom desempenho para soluções com tolerâncias maiores.

Já no cenário 2, apresentado na Figura 2, com apenas as restrições no sinal de saída, o incremento de controle computado atinge valores mais altos (Figura 2a). Pode-se também perceber uma pequena variação dos valores do sinal de controle computado entre os algoritmos, dentro da faixa de tolerância $\pm10^{-3}$. Na avaliação de desempenho apresentada na Figura 2b, com um número maior de restrições e um tipo de restrição mais complexa, todos os algoritmos gastaram um tempo de cômputo ligeiramente maior, entretanto o PFQP apresentou uma leve melhora ficando mais próximo do ADMM. O GPAD ainda se apresenta como a solução mais rápida, beneficiando-se da formulação dual que deixa a restrição mais simples antes de ser computada. Os algoritmos IP e Gurobi apresentaram pouca variação entre os cenários 1 e 2.

No terceiro cenário (Figura 3), com ambas as restrições e com a tolerância de $\pm10^{-6}$, a diferença no sinal de controle entre os algoritmos é imperceptível na escala apresentada no gráfico (Figura 3a). Na avaliação de tempo de cômputo apresentado na Figura 2b, os algoritmos de primeira ordem apresentaram um aumento considerável no número de iterações e consequentemente no tempo de cômputo. No pior caso, o ADMM teve um desempenho muito próximo dos métodos de barreira IP e Gurobi. O algoritmo PFQP teve um desempenho bem inferior ficando até 2 ordens de grandeza mais lento que o Gurobi. Por outro lado, o GPAD ainda se mostrou o algoritmo mais rápido. Neste cenário fica evidenciado que a utilização do método IP se torna competitivo com o aumento da exigência na tolerância da resposta. A vantagem da utilização do PFQP com relação ao aumento de restrições foi encoberto devido à sensibilidade do método ao aumento da exigência da tolerância.

Na Tabela 1 os tempos máximos de cômputo dos algoritmos, em milissegundos, são apresentados para cada cenário. O algoritmo GPAD foi o mais rápido em todos os cenários e, apesar de ser 8,72 vezes mais lento no terceiro cenário em relação ao primeiro, ainda se mostrou 6,29 vezes mais rápido que o IP. O ADMM e o PFQP apresentaram um tempo de cômputo 41,03 e 97,02 vezes mais lentos entre o cenário 3 e o cenário 1, respectivamente. Esse comportamento evidencia a sensibilidade destes algoritmos em relação às tolerâncias mais apertadas. Por outro lado, os algoritmos IP e Gurobi apresentaram aumentos mais modestos de 2,84 e 1,11 vezes, respectivamente, no terceiro cenário.

Tabela 1. Tempos máximos de cômputo em milissegundos.

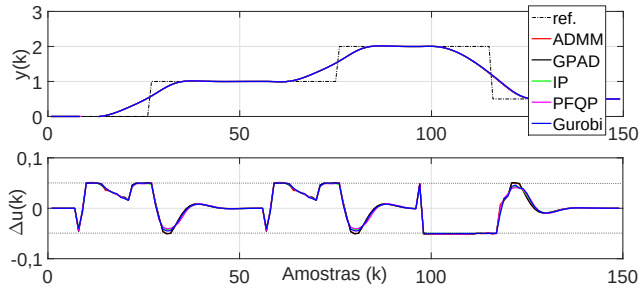
	Cenário 1	Cenário 2	Cenário 3
GPAD	3,31	6,16	28,87
ADMM	15,10	17,08	619,61
IP	63,92	98,00	181,46
PFQP	120,77	102,73	11717,00
Gurobi	242,86	259,31	269,61

5. CONCLUSÃO

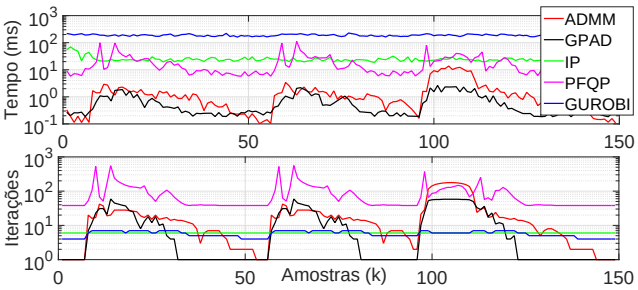
Foram apresentados algoritmos GPC com os métodos de barreira e PFQP. Uma comparação de desempenho dos algoritmos com o GPCGPAD e GPCADMM foi feita em um estudo de caso simulado em MATLAB. A partir do comportamento dos sinais em simulação, conclui-se que todos os algoritmos poderiam ser soluções plausíveis para o estudo de caso proposto, pois todos apresentaram um comportamento equivalente na saída da planta, mesmo com as tolerâncias mais baixas e uma menor precisão nos algoritmos de primeira ordem. O algoritmo GPCIP proposto, apesar de ser mais lento, torna-se uma boa opção para garantir o determinismo no pior caso do tempo de cômputo com baixa variabilidade nos momentos de restrição ativa. Outra vantagem evidenciada, apresenta-se nos casos que requerem uma maior precisão na solução. O algoritmo GPCPFQP proposto apresentou um melhor desempenho com a restrição na variável de saída, entretanto apresentou uma grande sensibilidade ao aumento da exigência na tolerância. Esse algoritmo pode ser uma boa opção em casos de baixa precisão e restrições complexas. A partir da comparação, pode-se concluir que os algoritmos estudados têm potencial para compor um conjunto de soluções com características distintas para a implementação dos controladores GPC. Dessa forma, o projetista pode escolher o algoritmo de acordo com os requisitos de projeto do controlador e aumentar a aplicação dos controladores preditivos com restrições em plantas com dinâmicas rápidas.

REFERÊNCIAS

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

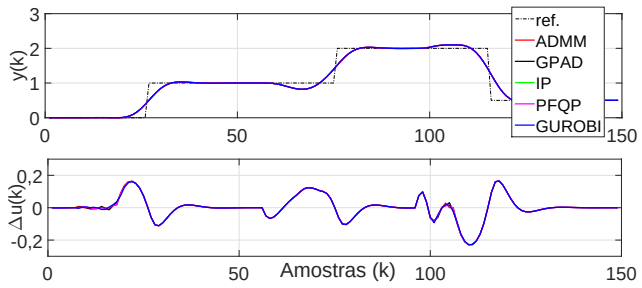


(a) Sinais de saída e incrementos de controle

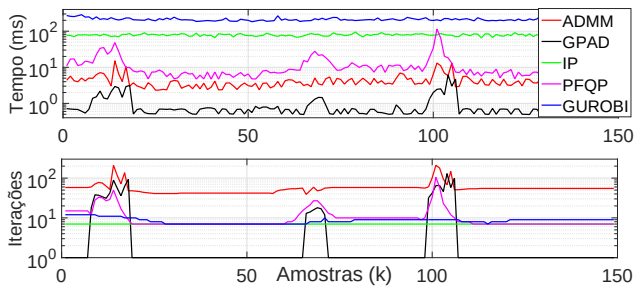


(b) Tempos de cômputo e do número de iterações

Figura 1. Comparação dos algoritmos no Cenário 1.

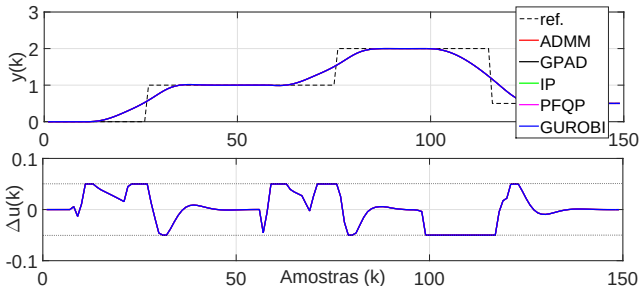


(a) Sinais de saída e incrementos de controle

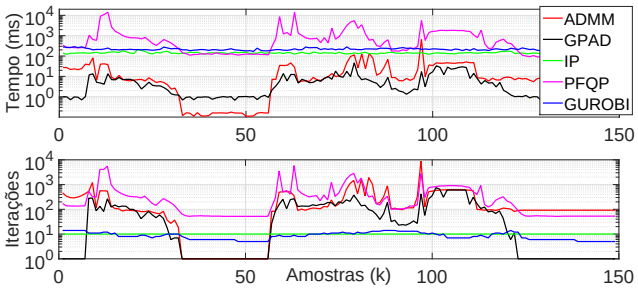


(b) Tempos de cômputo e do número de iterações

Figura 2. Comparação dos algoritmos no Cenário 2.



(a) Sinais de saída e incrementos de controle



(b) Tempos de cômputo e do número de iterações

Figura 3. Comparação dos algoritmos no Cenário 3.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1), 1–122.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge Univ. Press, Cambridge, MA.

Cairano, S.D., Brand, M., and Bortoff, S.A. (2013). Projection-free parallel quadratic programming for linear model predictive control. *Int. J. Control*, 86(8), 1367–1385.

Camacho, E. and Bordons, C. (2004). *Model Predictive Control*. Advanced Textbooks in Control and Signal Processing. Springer, London.

Cimini, G. and Bemporad, A. (2017). Exact complexity certification of active-set methods for quadratic programming. *IEEE Trans. Autom. Control*, 62(12), 6094–6109.

Clarke, D.W., Mohtadi, C., and Tuffs, P.S. (1987). Generalized predictive control - part 1: The basic algorithm. *Automatica*, 23(2), 137–148.

Ferreau, H., Almér, S., Verschueren, R., Diehl, M., Frick, D., Domahidi, A., Jerez, J., Stathopoulos, G., and Jones, C. (2017). Embedded optimization methods for industrial automatic control. In *Proc. 20th IFAC World Congr.* Toulouse, France.

Golub, G.H. and Van Loan, C.F. (1996). *Matrix Computations*. The Johns Hopkins University Press, third edition.

Gurobi (2020). *Gurobi Optimizer Reference Manual*. Inc. Gurobi Optimization.

Herceg, M., Jones, C.N., and Morari, M. (2015). Dominant speed factors of active set methods for fast MPC. *Optim. Contr. Appl. Met.*, 36(5), 608–627.

Kvasnica, M., Takács, B., Holaza, J., and Di Cairano, S. (2015). On region-free explicit model predictive control. In *Proc. 54th IEEE Conf. on Decision and Control (CDC)*, 3669–3674.

Normey-Rico, J.E. and Camacho, E.F. (2007). *Control of Dead-time Processes*. Springer, London.

O'Donoghue, B., Stathopoulos, G., and Boyd, S. (2013). A splitting method for optimal control. *IEEE Trans.*

Control Syst. Technol., 21(6), 2432–2442.

Patrinos, P. and Bemporad, A. (2014). An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Trans. Autom. Control*, 59(1), 18–33.

Peccin, V.B., Lima, D.M., Flesch, R.C.C., and Normey-Rico, J.E. (2019). Fast generalized predictive control based on accelerated dual gradient projection method. In *Proc. 12th IFAC Symp. on Dynamics and Control of Process Systems, including Biosystems (DYCOPS)*, 474–479. Florianópolis,Brazil.

Peccin, V.B., Lima, D.M., Flesch, R.C.C., and Normey-Rico, J.E. (2020). Fast constrained generalized predictive control with ADMM embedded in an FPGA. *IEEE Latin America Trans.*, 18(2), 422–429.

Pu, Y., Zeilinger, M.N., and Jones, C.N. (2017). Complexity certification of the fast alternating minimization algorithm for linear MPC. *IEEE Trans. Autom. Control*, 62(2), 888–893.

Roldao-Lopes, A., Shahzad, A., Constantinides, G.A., and Kerrigan, E.C. (2009). More flops or more precision? Accuracy parameterizable linear equation solvers for model predictive control. In *Proc. 17th IEEE Symp. Field Progr. Custom Comp. Machines*, 209–216.

Wills, A., Mills, A., and Ninness, B. (2011). FPGA implementation of an interior-point solution for linear model predictive control. In *Proc. 18th IFAC World Congr.* Milano, Italy.