

## Control por matriz dinámica rápido utilizando optimización en línea

Vinicius B. Peccin<sup>a</sup>, Daniel M. Lima<sup>b</sup>, Rodolfo C. C. Flesch<sup>c</sup>, Julio E. Normey-Rico<sup>c,\*</sup>

<sup>a</sup>Coordenação de Engenharia de Controle e Automação, Instituto Federal de Santa Catarina, 89813-000, Chapecó, SC, Brasil.

<sup>b</sup>Departamento de Controle, Automação e Computação, Universidade Federal de Santa Catarina, 89036-002, Blumenau, SC, Brasil. <sup>c</sup>Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina, 88040-900, Florianópolis, SC, Brasil.

**To cite this article:** Peccin, V. B., Lima, D. M., Flesch, R. C. C., Normey-Rico, J. E. 2022. Fast constrained dynamic matrix control algorithm with online optimization. Revista Iberoamericana de Automática e Informática Industrial 19, 330-342. <https://doi.org/10.4995/riai.2022.16619>

### Resumen

Este trabajo propone una técnica de control predictivo para ser aplicada en procesos rápidos utilizando optimización en línea. Actualmente, en el sector industrial, los controladores avanzados son cada vez más necesarios en los bajos niveles de automatización, que están asociados con tiempos de muestreo del orden de milisegundos o microsegundos. La programación cuadrática resultante del problema de control de un algoritmo de control predictivo con restricciones, como por ejemplo el control por matriz dinámica (en inglés *Dynamic Matrix Control* - DMC), que es uno de los más usados en la industria, se puede considerar computacionalmente costosa y se convierte en una limitación para empotrar y usar el DMC en plantas con tasas de muestreo rápidas. Este artículo propone una solución a este problema basada en el método de proyección de gradiente acelerado dual, que tiene tiempos de convergencia menores que otras soluciones de la literatura basadas en estrategias de control predictivo con optimización en línea. Como caso de estudio, se aplica el control propuesto a un sistema de suspensión semi-activa de un coche, que es un sistema con dinámicas muy rápidas, y se muestra que se puede controlar el proceso con tiempos de muestreo de 5 ms. Además, el algoritmo desarrollado se implementó en una matriz de puertos programables (en inglés *field-programmable gate array* – FPGA) y el problema de programación cuadrática resultante se calculó en microsegundos, lo que permite el uso del DMC en procesos muy rápidos.

*Palabras clave:* Control predictivo, Procesos rápidos, Optimización, FPGA, Sistemas automotores.

### Fast constrained dynamic matrix control algorithm with online optimization

#### Abstract

This work proposes a predictive control technique to be applied in fast processes using online optimization. Currently, advanced controllers are increasingly needed in industry at low levels of automation, which are associated with sampling times on the order of milliseconds or microseconds. The quadratic programming resulting from the control problem of a predictive control algorithm with constraints, such as Dynamic Matrix Control (DMC), which is one of the most used alternatives in industry, can be considered computationally expensive and becomes a limitation to embed and use the DMC in plants with fast sample rates. This paper proposes a solution to this problem based on the dual accelerated gradient projection method, which has shorter convergence times than other solutions in the literature based on predictive control strategies with online optimization. The proposed approach was tested in simulation to control a semiactive automotive suspension system, which has fast dynamics, showing that satisfactory results can be obtained with a sampling time of 5 ms. Moreover, the proposed control was implemented in a field-programmable gate array (FPGA) and the resulting quadratic programming problem was calculated in microseconds, which allows the use of the DMC to control very fast processes.

*Keywords:* Model Predictive Control, Optimization, Fast Processes, FPGA, Automotive systems.

\*Autor para correspondencia: julio.normey@ufsc.br

## 1. Introducción

El control por matriz dinámica (DMC, en inglés *Dynamic Matrix Control*) fue propuesto por Cutler and Ramaker (1980) y se ha convertido en uno de los métodos de control avanzados más populares en la industria. Tiene un uso generalizado en la industria petroquímica, especialmente en columnas de destilación, pero también se utiliza en otros sectores industriales (Camacho and Bordons, 2004). El DMC es una formulación particular de control predictivo que utiliza un modelo de respuesta al escalón para el cálculo de las predicciones de la salida del proceso. La investigación sobre aplicaciones industriales del DMC aún está activa y algunos trabajos recientes pueden citarse. El trabajo de Fernandes et al. (2020), utiliza el DMC en una planta que integra un proceso de separación de aire con un ciclo de Allam. En el trabajo de He and Lima (2019) una estrategia con base en un DMC se usa para controlar una central eléctrica de carbón. Y en Wang et al. (2020) el DMC se utiliza en el contexto del control predictivo distribuido, aplicado a un proceso de reacción y separación.

A pesar de ser masivamente usado en el contexto industrial, casi no hay aplicaciones del DMC en procesos que usen frecuencias de muestreo rápidas. En la literatura, la investigación sobre algoritmos MPC de computación rápida se centra principalmente en formulaciones de MPC en espacio de estados con diferentes enfoques para resolver el problema de optimización resultante. Sin embargo, estas soluciones en el espacio de estados no se utilizan mucho en aplicaciones industriales, donde las estrategias MPC basadas en modelos de entrada/salida tienen mucha más aceptación. En general, el tiempo que se tarda en resolver el problema de programación cuadrática (QP, del inglés *Quadratic Programming*) para obtener la acción de control en cada iteración es el principal obstáculo para acelerar el cálculo del MPC. Por lo tanto, la elección del método de resolución del MPC-QP es fundamental para obtener un algoritmo eficiente. Un enfoque es reescribir el QP como un problema de programación paramétrica y luego calcular una solución *offline* basada en regiones. Esta estrategia, que se denomina en la literatura como MPC explícito (Pistikopoulos et al., 2015), tiene como inconveniente que el número de regiones crece muy rápidamente con el número de estados del sistema y, en consecuencia, aumenta la memoria necesaria para almacenar los datos. En Borrelli et al. (2010) y Kvasnica et al. (2015), se presentan soluciones de MPC en espacio de estados explícitas sin regiones, para mitigar este problema, donde se evita la generación de regiones críticas y se las reemplaza por la enumeración de conjuntos activos óptimos, lo que reduce la memoria requerida para el almacenamiento. Sin embargo, su uso todavía está limitado a problemas con pocas restricciones ya que el número de posibles combinaciones de restricciones activas aumenta exponencialmente con el número de restricciones del problema (Ahmadi-Moshkenani et al., 2018). Entre las soluciones informáticas *online*, el método de los conjuntos activos (Cimini and Bemporad, 2017); (Herceg et al., 2015), el método del punto interior (Roldao-Lopes et al., 2009); (Wills et al., 2011) y los métodos de primer orden merecen ser resaltados. Entre los métodos de primer orden se pueden destacar algunos trabajos. En el artículo de Patrinos and Bemporad (2014) se presenta una solución basada en el método de gradiente proyectado acelerado dual (GPAD) para la formulación dual del QP del MPC.

En el trabajo de O'Donoghue et al. (2013) se presenta una solución a través del método de operador de partición, llamado de método de dirección alterna de multiplicadores (ADMM) y en el trabajo de Pu et al. (2017) los autores utilizan un método de operador de partición llamado algoritmo de minimización de alternancia rápida (FAMA). FAMA puede verse como una versión acelerada equivalente a ejecutar el algoritmo iterativo rápido de contracción del límite (FISTA) en el problema dual (Goldstein et al., 2014). Otro enfoque con base en métodos de primer orden se presenta en Cairano et al. (2013), que utiliza un algoritmo de optimización sin proyección basado en el método de programación cuadrática paralela. Lo interesante de los métodos de primer orden es que requieren pocos recursos computacionales y son relativamente simples de implementar en comparación con los enfoques tradicionales de optimización en línea. Sin embargo, una desventaja común de los métodos de primer orden es que las prestaciones en términos numéricos dependen de las características del problema a resolver (Ferreau et al., 2017).

Una ventaja de usar el DMC y no las formulaciones de espacio de estados es el número reducido de variables para modelar el QP resultante, especialmente para modelos con muchos estados. Además, una aplicación con DMC requiere un número reducido de sensores y no necesita observadores de estado. Matemáticamente, siempre es posible reescribir el DMC en una formulación de espacio de estados, sin embargo, debido al uso del modelo de respuesta al escalón, el número de estados del modelo que se obtiene es muy grande comparado al número de entradas y salidas. Por ejemplo, un sistema de una entrada y una salida representado de esta manera tendría tantos estados como coeficientes de la respuesta al escalón (Lee et al., 1994). Para un sistema de múltiples entradas y múltiples salidas, el número de estados hace que la implementación del DMC en el espacio de estados no sea práctica para sistemas rápidos.

Aunque la mayoría de los algoritmos de computación rápida para MPC con restricciones utilizan exclusivamente formulaciones de espacio de estados, algunos trabajos recientes presentan soluciones con enfoques prácticos para incorporar implementaciones de MPC de computación rápida, basadas en estrategias de MPC con modelos de entrada/salida. Por ejemplo, en Wojtulewicz and Ławryńczuk (2018) los autores presentan una implementación de DMC en una FPGA, sin embargo, el estudio solo analiza el caso sin restricciones, que no requiere que un QP se resuelva en línea. En Peccin et al. (2019), se presenta una implementación de otra forma de controlador MPC, llamada control predictivo generalizado (GPC), que utiliza un modelo de entrada/salida de función de transferencia de tiempo discreto. En ese trabajo se propuso un algoritmo de cómputo rápido para GPC con restricciones y se utilizó el método GPAD.

Con el fin de contribuir a las aplicaciones prácticas de DMC en procesos con dinámica rápida, el objetivo principal de este trabajo es proponer un algoritmo DMC de computación rápida para el caso con restricciones, que hace uso del método de gradiente proyectado acelerado dual (GPAD) para resolver el QP resultante en cada período de muestreo. El algoritmo propuesto se llama DMCGPAD y se puede utilizar para controlar sistemas multivariados con dinámica rápida. Además, es adecuado para aplicaciones empotradas, ya que cumple con algunos requisitos para este tipo de aplicaciones, como la simplicidad para gene-

rar el código y la necesidad de poco espacio de memoria para almacenar los datos que definen el problema de optimización y el código que implementa la solución. Por tanto, las tres aportaciones principales de este artículo son las siguientes:

- la propuesta de un algoritmo DMC multivariable con restricciones eficiente, diseñado para requerir recursos computacionales modestos;
- un estudio comparativo del algoritmo propuesto con otros métodos de cálculo rápido utilizando el problema de control de una suspensión semi-ativa de un coche;
- la presentación de resultados experimentales con los algoritmos integrados en una FPGA.

El resto de este artículo está organizado de la siguiente manera: el Apartado 2 revisa el algoritmo DMC multivariable y algunos detalles de formulación que se exploran para mejorar su implementación. El algoritmo DMC propuesto con el método de gradiente proyectado acelerado dual se presenta en el Apartado 3. En el Apartado 4 se estudia la aplicación del DMCG-PAD en el control de la suspensión semi-activa de un coche, y se compara el algoritmo propuesto con otras estrategias de control. En el Apartado 5 se presenta una implementación de FPGA y se analizan los requisitos de tiempo de cálculo. Finalmente, el Apartado 6 es dedicado a las conclusiones del artículo.

## 2. Control por Matriz Dinámica

La función objetivo del DMC, para el caso monovariable, viene dada por (Camacho and Bordons, 2004):

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

donde  $\hat{y}(k+j|k)$  representa la predicción de la salida del proceso en el instante futuro de tiempo discreto  $k+j$ , determinada en  $k$ ;  $w(k+j)$  es la trayectoria de referencia en  $k+j$ ;  $\Delta u$  es el incremento de control;  $\delta(j)$  y  $\lambda(j)$  son las secuencias de ponderación del error y del esfuerzo de control, respectivamente.  $N = N_2 - N_1 + 1$  define el horizonte de predicción donde  $N_1$  es el comienzo del horizonte, generalmente definido como igual a  $d+1$ , siendo  $d$  el tiempo muerto del sistema, y  $N_2$  es el final del horizonte;  $N_u$  es el horizonte de control.

El modelo utilizado por el DMC para predecir las salidas del sistema, que debe ser estable en bucle abierto, se basa en la respuesta al escalón. La respuesta al escalón de la planta con condiciones iniciales cero y sin perturbaciones viene dada por:

$$y_0(k) = \sum_{i=1}^{\infty} g_i \Delta u(k-i), \tag{2}$$

donde  $g_i, i = 1, 2, \dots$ , representan los coeficientes de la respuesta al escalón de bucle abierto.

Usando el modelo de (2), el DMC calcula las predicciones de la salida del sistema con:

$$\hat{y}(k+j|k) = \sum_{i=1}^{\infty} g_i \Delta u(k+j-i) + \hat{n}(k+j|k), \tag{3}$$

donde  $\hat{n}(k+j|k)$  representa la predicción de la perturbación en  $k+j$  obtenida en  $k$  (Camacho and Bordons, 2004). La predicción de la salida puede ser dividida en dos términos, el primero con las acciones de control futuras que serán calculadas y el segundo con los valores de las acciones de control pasadas. Así, (3) puede ser reescrita como:

$$\hat{y}(k+j|k) = \sum_{i=1}^j g_i \Delta u(k+j-i) + \sum_{i=j+1}^{\infty} g_i \Delta u(k+j-i) + \hat{n}(k+j|k). \tag{4}$$

Como los valores futuros de las perturbaciones son desconocidos, se los puede considerar constantes e iguales al valor actual, o sea,  $\hat{n}(k+j|k) = \hat{n}(k|k) = y(k) - y_0(k|k)$ . Utilizando (2), (4) se puede representar como:

$$\hat{y}(k+j|k) = \sum_{i=1}^j g_i \Delta u(k+j-i) + \sum_{i=j+1}^{\infty} g_i \Delta u(k+j-i) + y(k) - \sum_{i=1}^{\infty} g_i \Delta u(k-i), \tag{5}$$

y puede ser reescrita como:

$$\hat{y}(k+j|k) = \sum_{i=1}^j g_i \Delta u(k+j-i) + f(k+j), \tag{6}$$

siendo  $f(k+j)$  la respuesta libre del sistema, que puede ser calculada de manera simple agrupando las sumas y truncándolas en  $i = N_{ss}$ . Esto es posible pues el modelo de bucle abierto es estable, así  $g_i \approx g_{N_{ss}}$  para  $i > N_{ss}$ , por lo que, en la práctica, apenas guardando  $N_{ss}$  coeficientes se tiene una buena representación del comportamiento del proceso.  $N_{ss}$  se define como el "horizonte del modelo". Así,  $N_{ss}$  solamente se usa para describir el modelo del proceso, y no es igual al horizonte de predicción  $N$ , usado para el cálculo de las predicciones que afectan la función objetivo  $J$ . El valor de  $N_{ss}$  se puede estimar aproximadamente como el tiempo de que el sistema lleva para llegar al estado estacionario en un ensayo de respuesta al escalón dividido por el período de muestreo.

Representando (6) en forma matricial, la predicción futura de la salida del sistema puede ser dividida en dos términos, la respuesta libre y la respuesta forzada, modeladas por  $\mathbf{f}$  y  $\mathbf{G}\Delta\mathbf{u}$ , respectivamente:

$$\hat{\mathbf{y}} = \mathbf{G}\Delta\mathbf{u} + \mathbf{f}, \tag{7}$$

con  $\hat{\mathbf{y}} \in \mathbb{R}^N, \Delta\mathbf{u} \in \mathbb{R}^{N_u}, \mathbf{G} \in \mathbb{R}^{N \times N_u}, \mathbf{f} \in \mathbb{R}^N$  dados por:

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}(k+N_1|k) \\ \hat{y}(k+N_1+1|k) \\ \vdots \\ \hat{y}(k+N_2|k) \end{bmatrix}, \quad \Delta\mathbf{u} = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_u-1) \end{bmatrix},$$

$$\mathbf{G} = \begin{bmatrix} g_{N_1} & g_{N_1-1} & \cdots & g_{N_1-N_u+1} \\ g_{N_1+1} & g_{N_1} & \cdots & g_{N_1-N_u+2} \\ \vdots & \vdots & \vdots & \vdots \\ g_{N_2} & g_{N_2-1} & \cdots & g_{N_2-N_u+1} \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_{N_1} \\ f_{N_1+1} \\ \vdots \\ f_{N_2} \end{bmatrix}.$$

La matriz  $\mathbf{G}$  puede ser calculada directamente dado que sus columnas son formadas por los coeficientes de la respuesta al escalón de la planta en bucle abierto.

Para calcular la respuesta libre del sistema más fácilmente, se puede utilizar un enfoque recursivo. Al comienzo de cada período de muestreo hay un vector de predicción de la salida del sistema del paso anterior,

$$\hat{\mathbf{f}}(k-1) = [\hat{y}(k|k-1) \hat{y}(k+1|k-1) \dots \hat{y}(k+N_{ss}-1|k-1)]^T.$$

El vector actualizado se calcula con:

$$\hat{\mathbf{f}}(k) = \hat{\mathbf{f}}(k-1) + \begin{bmatrix} g_1 \\ \vdots \\ g_{N_{ss}} \end{bmatrix} \Delta u(k-1). \quad (8)$$

El primer elemento de  $\hat{\mathbf{f}}$  es  $\hat{y}(k|k)$ , que se usa para calcular el error de predicción  $e(k|k)$ . Este valor luego se descarta del vector ya que ahora solo se necesitan las predicciones desde  $k+1$  hasta  $k+N_{ss}$ . Suponiendo que la planta se inicia en estado estacionario,  $\hat{\mathbf{f}}(0)$  tiene todos sus elementos iguales a  $y(0)$ . El vector  $\hat{\mathbf{f}}$  también debe cambiarse en cada iteración del control debido al horizonte deslizante. Sin embargo, cuando el vector se desplaza, el elemento  $\hat{y}(k|k)$  se descarta y, para mantener las dimensiones adecuadas, el valor de  $\hat{y}(k+N_{ss}-1|k)$  es repetido en la última posición del vector. Este enfoque se puede adoptar porque, para sistemas estables,  $\hat{y}(k+N_{ss}-1|k) \approx \hat{y}(k+N_{ss}|k)$  para un  $N_{ss}$  suficientemente grande (Lima et al., 2014).

El siguiente paso para el cálculo del control es aplicar las secuencias de predicción de salida (7) a la función de costo del DMC (1), que se puede reescribir como:

$$h(\Delta \mathbf{u}) = \frac{1}{2} \Delta \mathbf{u}^T \mathbf{H} \Delta \mathbf{u} + \mathbf{b}^T \Delta \mathbf{u}, \quad (9)$$

donde

$$\mathbf{H} = 2(\mathbf{G}^T \mathbf{Q}_\delta \mathbf{G} + \mathbf{Q}_l),$$

$$\mathbf{b}^T = 2(\mathbf{f} - \mathbf{w})^T \mathbf{Q}_\delta \mathbf{G},$$

$\mathbf{Q}_\delta = \text{diag}(\delta_j) \in \mathbb{R}^{N \times N}$  y  $\mathbf{Q}_l = \text{diag}(\lambda_j) \in \mathbb{R}^{N_u \times N_u}$  son matrices de ponderación diagonales definidas positivas, y

$$\mathbf{w} = [w(k+N_1) \ w(k+N_1+1) \dots \ w(k+N_2)]^T$$

es el vector de trayectoria de referencia.

Finalmente, el vector de incremento de control  $\Delta \mathbf{u}$  se puede obtener, en cada iteración de control, a partir de la solución del siguiente QP:

$$\begin{aligned} \min_{\Delta \mathbf{u}} \quad & h(\Delta \mathbf{u}) \\ \text{s.a.} \quad & \bar{\mathbf{R}} \Delta \mathbf{u} \leq \bar{\mathbf{r}}, \end{aligned} \quad (10)$$

con la matriz de restricción  $\bar{\mathbf{R}} \in \mathbb{R}^{n_{rin} \times N_u}$ , el vector  $\bar{\mathbf{r}} \in \mathbb{R}^{n_{rin}}$  y  $n_{rin}$  que representan el número de restricciones de desigualdad. La forma condensada de representar las restricciones  $\bar{\mathbf{R}} \Delta \mathbf{u} \leq \bar{\mathbf{r}}$  es genérica y permite la representación de muchos tipos de restricciones, como magnitud, tasa de cambio y sobrepaso para las variables de entrada y salida (Camacho and Bordons, 2004).

La generalización a los sistemas multivariables es sencilla. Para representar las predicciones de  $n_o$  salidas, los vectores asociados con cada salida se pueden agrupar en un solo vector

$\hat{\mathbf{y}}_m \in \mathbb{R}^{\sum_{p=1}^{n_o} N_2^{(p)} - N_1^{(p)} + 1}$  y las  $n_i$  señales de control en un vector  $\mathbf{u}_m \in \mathbb{R}^{\sum_{i=1}^{n_i} N_u^{(i)}}$ , tal que:

$$\hat{\mathbf{y}}_m = [\hat{\mathbf{y}}^{(1)T} \dots \hat{\mathbf{y}}^{(n_o)T}]^T, \quad \mathbf{u}_m = [\mathbf{u}^{(1)T} \dots \mathbf{u}^{(n_i)T}]^T, \quad (11)$$

con  $N_1^{(p)}$ ,  $N_2^{(p)}$  los horizontes de la salida  $p$  y  $N_u^{(i)}$  el horizonte del control  $l$ .

El modelo de predicción multivariable del DMC sigue los mismos pasos desarrollados para el caso monovariable y se puede escribir como en (7). En el caso multivariable, la matriz  $\mathbf{G}_m$  está formada por otras matrices  $\mathbf{G}^{(p,l)}$ , donde  $\mathbf{G}^{(p,l)}$  es una matriz de coeficientes de respuesta al escalón que relaciona la salida  $p$  con la entrada  $l$ . Por tanto, la matriz  $\mathbf{G}_m$  viene dada por:

$$\mathbf{G}_m = \begin{bmatrix} \mathbf{G}^{(1,1)} & \dots & \mathbf{G}^{(1,n_i)} \\ \mathbf{G}^{(2,1)} & \dots & \mathbf{G}^{(2,n_i)} \\ \vdots & \ddots & \vdots \\ \mathbf{G}^{(n_o,1)} & \dots & \mathbf{G}^{(n_o,n_i)} \end{bmatrix}.$$

Para obtener el vector de respuesta libre generalizado  $\mathbf{f}_m$ , las respuestas libres para cada salida del sistema se agrupan como:

$$\mathbf{f}_m = [\mathbf{f}^{(1)T} \dots \mathbf{f}^{(n_o)T}]^T.$$

Siguiendo la misma lógica para obtener la ley de control para el caso monovariable y generalizando  $\mathbf{w}_m$  con la agrupación de referencias futuras para cada salida predicha de manera similar a (11), el problema de optimización multivariable se puede representar de la misma forma estándar (10).

### 3. Algoritmo DMCGPAD

En este apartado se presenta un algoritmo para el DMC con el método de gradiente proyectado acelerado dual. El GPAD fue propuesto por Patrinos and Bemporad (2014) y usa la forma dual del QP con el método de aceleración propuesto por Nesterov (1983). El GPAD es adecuado para un MPC computacionalmente eficiente porque es simple, es fácil de codificar y su costo computacional crece linealmente con el horizonte de control. Además, permite una estimación del número máximo de iteraciones para obtener el valor óptimo para una determinada precisión deseada.

El paso principal del método de gradiente proyectado es la solución del problema sin restricciones en la dirección descendente seguida de la proyección sobre el conjunto de restricciones activas,

$$\mathbf{x}_{j+1} = P(\mathbf{x}_j - l_j \nabla h(\mathbf{x}_j)), \quad (12)$$

donde  $P$  es una función de proyección ortogonal,  $l_j > 0$  es el tamaño del paso y  $-\nabla h$  representa el anti-gradiente de la función  $h$ .

El GPAD usa la forma dual del problema QP para simplificar la función de proyección. El dual de la función (9) viene dado por:

$$h_d(\psi) = -\left( \frac{1}{2} \psi^T \bar{\mathbf{R}} \mathbf{H}^{-1} \bar{\mathbf{R}}^T \psi + (\bar{\mathbf{R}} \mathbf{H}^{-1} \mathbf{b} + \bar{\mathbf{r}})^T \psi + \frac{1}{2} \mathbf{b}^T \mathbf{H}^{-1} \mathbf{b} \right), \quad (13)$$

donde  $\psi \in \mathbb{R}^{n_{rin}}$  es el vector de multiplicadores lagrangianos asociados con las restricciones de desigualdad y la función lagrangiana es:

$$\mathcal{L}(\Delta \mathbf{u}, \psi) = \frac{1}{2} \Delta \mathbf{u}^T \mathbf{H} \Delta \mathbf{u} + \mathbf{b}^T \Delta \mathbf{u} + \psi^T (\bar{\mathbf{R}} \Delta \mathbf{u} - \bar{\mathbf{r}}).$$

El QP dual de (10) puede ser representado como:

$$\begin{aligned} \min_{\psi} \quad & -h_d(\psi) \\ \text{s.a.} \quad & \psi \geq \mathbf{0}. \end{aligned} \quad (14)$$

El paso principal de GPAD (12) para el problema de doble QP (14) se convierte en:

$$\psi_{j+1} = \max \left( \psi_j - \frac{1}{L} (\bar{\mathbf{R}}\mathbf{H}^{-1}\bar{\mathbf{R}}^T\psi_j + \bar{\mathbf{R}}\mathbf{H}^{-1}\mathbf{b} + \bar{\mathbf{r}}), \mathbf{0} \right), \quad (15)$$

donde el operador máx, definido para ser elemento por elemento, es la función de proyección,  $\mathbf{0}$  es un vector con todos los elementos iguales a cero y dimensiones apropiadas, y  $1/L$  representa el tamaño del paso.  $L$  es una constante de Lipschitz y se puede obtener a partir de la formación de la matriz hessiana de (13) y el cálculo de su norma espectral,  $L = \|\bar{\mathbf{R}}\mathbf{H}^{-1}\bar{\mathbf{R}}^T\|_2$ .

La ecuación (15) puede ser dividida en dos pasos:

$$\begin{aligned} \Delta\mathbf{u}_j &= -\mathbf{H}^{-1}(\mathbf{b} + \bar{\mathbf{R}}^T\psi_j), \\ \psi_{j+1} &= \max \left( \psi_j + \frac{1}{L}(\bar{\mathbf{R}}\Delta\mathbf{u}_j + \bar{\mathbf{r}}), \mathbf{0} \right). \end{aligned} \quad (16)$$

El paso de aceleración GPAD, basado en el método de Nesterov, se realiza con una operación a priori, que se inserta en el cálculo de  $\psi$ :

$$\mathbf{a} = \psi_j + \beta(\psi_j - \psi_{j-1}), \quad (17)$$

donde  $\beta$  se puede calcular simplemente como  $\beta = \frac{j-1}{j+2}$ , y garantiza la convergencia del algoritmo (Patrinos and Bemporad, 2014).

Las propiedades de factibilidad y optimalidad primaria se utilizaron como criterios de detención para los algoritmos. Para la viabilidad primaria, es posible probar si las restricciones se cumplen dentro de la tolerancia deseada  $\epsilon_f$ , es decir:

$$\bar{\mathbf{R}}\Delta\mathbf{u}_j - \bar{\mathbf{r}} \leq \epsilon_f. \quad (18)$$

Para el criterio de optimalidad primaria, se puede utilizar una propiedad derivada de la dualidad fuerte, que implica:

$$h(\Delta\mathbf{u}) - h(\Delta\mathbf{u}^*) \leq h(\Delta\mathbf{u}) - h_d(\psi), \quad (19)$$

donde  $h$  es la función de costo principal,  $h_d$  es la función de costo dual y  $\Delta\mathbf{u}^*$  representa el valor óptimo de  $\Delta\mathbf{u}$ . Por lo tanto, para obtener una solución  $\epsilon_o$  óptima, una condición necesaria es que:

$$h(\Delta\mathbf{u}) - h_d(\psi) \leq \epsilon_o. \quad (20)$$

Sustituyendo (9) y (13) en (20) y usando la definición de lagrangiano, tenemos:

$$-\psi^T(\bar{\mathbf{R}}\Delta\mathbf{u}_j - \bar{\mathbf{r}}) \leq \epsilon_o. \quad (21)$$

De los desarrollos presentados, DMCGPAD se puede resumir en el algoritmo 1, donde  $\mathbf{g}^{(p,d)}$  es el vector de coeficientes de respuesta al escalón del sistema de bucle abierto de la entrada  $l$  para la salida  $p$ . El símbolo  $\leftarrow$  denota la asociación de un valor a una variable,  $\lambda_{\max}(\mathbf{M})$  es el valor propio máximo de la

matriz  $\mathbf{M}$  y la notación  $\mathbf{V}_{n:m}$  representa un subvector compuesto por los elementos  $n$  a  $m$  del vector  $\mathbf{V}$ .  $j_{\max}$  es el número máximo de iteraciones de GPAD. Si se alcanza el número máximo de iteraciones del bucle de optimización, se utiliza el incremento de control subóptimo calculado. Se supone que el problema es factible, los casos no factibles no fueron abordados en este trabajo.

#### 4. Caso de estudio

El caso de estudio considerado fue el problema de control de un sistema de suspensión de un coche con amortiguador semiactivo electroreológico (ER) (Morato et al., 2021). Este caso de estudio se basa en el modelo a escala del proyecto INOVE<sup>1</sup>. En este vehículo, la suspensión de cada rueda es un sistema independiente cuyo objetivo es aumentar la comodidad y seguridad de los pasajeros a bordo. En la Figura 1, se presenta un diagrama esquemático del sistema de suspensión referido a la rueda del vehículo.

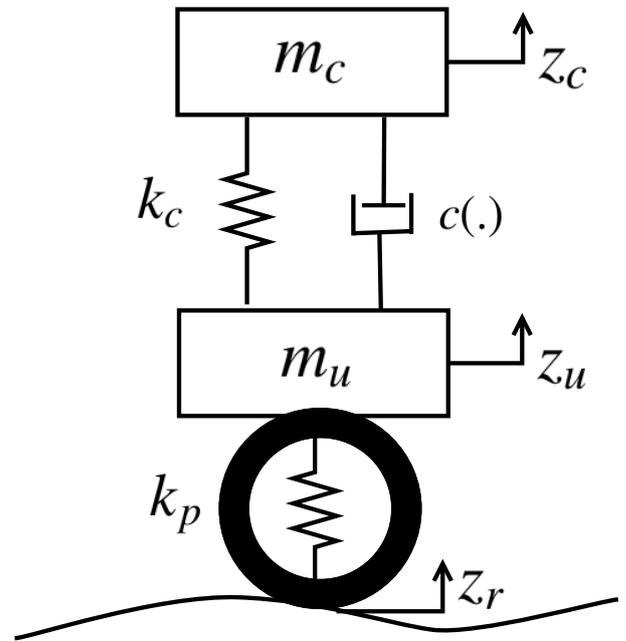


Figura 1: Diagrama esquemático del problema de control de la suspensión.

La posición vertical del chasis viene dada por  $z_c$ . Las posiciones verticales del centro de masa de la rueda están dadas por  $z_u$ . La variable  $z_r$  representa el perfil de la carretera, que en este caso se modelará como una perturbación en el sistema de control. La masa equivalente del vehículo conectada al amortiguador está representada por  $m_c$  y la masa del conjunto que acopla el amortiguador a la rueda por  $m_r$ . La rigidez del muelle del sistema de amortiguación viene dada por el coeficiente  $k_c$  y el neumático está modelado por un muelle con coeficiente  $k_p$ . El modelo dinámico linealizado de la fuerza del amortiguador semiactivo ER viene dado por:

$$F_d(t) = k_0(z_c(t) - z_u(t)) + c_0(\dot{z}_c(t) - \dot{z}_u(t)) + f_c u(t), \quad (22)$$

<sup>1</sup>Para más detalles vea [www.gipsa-lab.fr/projet/inove](http://www.gipsa-lab.fr/projet/inove).

**Algoritmo 1: MIMO DMCGPAD**

**Entrada:**  $\mathbf{g}^{(1,1)}, \dots, \mathbf{g}^{(n_o, n_i)}, \mathbf{R}, \bar{\mathbf{r}}, u^{(1)}(k-1), \dots, u^{(n_i)}(k-1)$

**Salida:**  $u^{(1)}(k), \dots, u^{(n_o)}(k)$

**Datos:**  $\mathbf{Q}_\lambda, \mathbf{Q}_\delta, \mathbf{N}_u, \mathbf{N}_1, \mathbf{N}_2, \mathbf{N}_{ss}, \psi_0 = \psi_{-1} = \mathbf{0}, \epsilon_f, \epsilon_o, j_{max}$

**inicio**

**para**  $p = 1 : n_o$  **hacer**

$N^{(p)} \leftarrow N_2^{(p)} - N_1^{(p)} + 1;$

**para**  $l = 1 : n_i$  **hacer**

**para**  $i = 1 : N_u^{(l)}$  **hacer**

$\mathbf{G}_{i:N^{(p)}, i}^{(p,l)} \leftarrow \mathbf{g}_{N_1^{(p)}:N_2^{(p)}+1-i}^{(p,l)};$

$\mathbf{G}_{mpl} \leftarrow \mathbf{G}^{(p,l)};$

$\hat{\mathbf{f}}^{(p)} \leftarrow \mathbf{1}_{N_{ss}^{(p)}} y^{(p)}(0);$

$\mathbf{H} \leftarrow (\mathbf{G}_m^T \mathbf{Q}_\delta \mathbf{G}_m + \mathbf{Q}_\lambda);$

$\check{\mathbf{H}} \leftarrow \mathbf{H}^{-1};$

$L \leftarrow \sqrt{\lambda_{max}((\check{\mathbf{R}}\check{\mathbf{H}}\check{\mathbf{R}}^T)^2)};$

$k \leftarrow 0;$

**mientras modo automático hacer**

**si período de muestreo entonces**

            Obtener las salidas  $y^{(1)}(k) \dots y^{(n_o)}(k);$

            Definir las referencias  $\mathbf{w}_m \leftarrow [\mathbf{w}^{(1)}(k) \dots \mathbf{w}^{(n_o)}(k)]^T;$

**para**  $p = 1 : n_o$  **hacer**

$\hat{\mathbf{f}}^{(p)} \leftarrow \hat{\mathbf{f}}^{(p)} + \mathbf{g}^{(p)} \Delta u^{(p)}(k-1);$

$\mathbf{e}^{(p)} \leftarrow (y^{(p)}(k) - \hat{f}_0^{(p)});$

$\hat{\mathbf{f}}^{(p)} \leftarrow [\hat{f}_1^{(p)} \dots \hat{f}_{N_{ss}^{(p)}-1}^{(p)} \hat{f}_{N_{ss}^{(p)}-1}^{(p)}]^{T};$

$\mathbf{f}^{(p)} \leftarrow \hat{\mathbf{f}}^{(p)}_{N_1^{(p)}:N_2^{(p)}} + \mathbf{1}_{N^{(p)} \times 1} \mathbf{e}^{(p)};$

$\mathbf{f}_m \leftarrow [\mathbf{f}^{(1)} \dots \mathbf{f}^{(n_o)}]^{T};$

$\mathbf{b}^T \leftarrow (\mathbf{f}_m - \mathbf{w}_m)^T \mathbf{Q}_\delta \mathbf{G}_m;$

**mientras**  $j < j_{max}$  **hacer**

**si**  $j = 0$  **entonces**

$\beta \leftarrow 0;$

**en otro caso**

$\beta \leftarrow \frac{j-1}{j+2};$

$\mathbf{a} \leftarrow \psi_j + \beta(\psi_j - \psi_{j-1});$

$\Delta \mathbf{u}_m \leftarrow -\check{\mathbf{H}}(\mathbf{R}^T \mathbf{a} + \mathbf{b});$

$\mathbf{s} \leftarrow (1/L)(\mathbf{R} \Delta \mathbf{u}_m - \bar{\mathbf{r}});$

**si**  $\mathbf{s}_i \leq \epsilon_f/L, \forall i = 1, \dots, n_{rin}$  **entonces**

**si**  $-\mathbf{a}^T \mathbf{s} < \epsilon_o/L$  **entonces**

                    rompe;

$\psi_{j+1} \leftarrow \text{máx}(\mathbf{a} + \mathbf{s}, \mathbf{0});$

$j \leftarrow j + 1;$

**para**  $l = 1 : n_i$  **hacer**

$u^{(l)}(k) \leftarrow u^{(l)}(k-1) + \Delta u^{(l)}(k);$

$k \leftarrow k + 1;$

donde  $k_0$  representa el coeficiente del muelle y  $c_0$  el coeficiente de rozamiento viscoso de la parte pasiva del amortiguador. El coeficiente de fuerza dinámica del fluido ER está dado por  $f_c$ . La señal  $u(t)$  representa el ciclo de trabajo de una señal PWM que regula el voltaje de entrada responsable de generar el campo eléctrico en el amortiguador ER. Este campo eléctrico varía la viscosidad del fluido ER. En la práctica, la señal PWM,  $0 \leq u(t) \leq 1$ , es la variable manipulada del sistema de control.

Las ecuaciones dinámicas relacionadas con el cuerpo del chasis y la rueda se pueden obtener usando las leyes del movimiento alrededor de un punto de equilibrio mediante:

$$\begin{aligned} m_c \ddot{z}_c &= -F_m(t) - F_a(t) \\ m_u \ddot{z}_u &= F_m(t) + F_a(t) - F_p(t), \end{aligned} \quad (23)$$

con

$$\begin{aligned} F_m(t) &= k_c(z_c(t) - z_u(t)) \\ F_p(t) &= k_p(\dot{z}_u(t) - \dot{z}_r(t)). \end{aligned} \quad (24)$$

El propósito principal de los controles es disminuir la aceleración del chasis ( $A_c(t) = \ddot{z}_c(t)$ ) y la aceleración de las ruedas ( $A_u(t) = \ddot{z}_u(t)$ ) que están directamente relacionadas con la comodidad del pasajero dentro del vehículo (Kiencke and Nielsen, 2000). Por tanto,  $A_c(t)$  y  $A_u(t)$  son las variables controladas consideradas. En la práctica, estas aceleraciones se pueden medir con acelerómetros.

Las funciones de transferencia del sistema se definen como:

$$\begin{bmatrix} A_c(s) \\ A_u(s) \end{bmatrix} = \begin{bmatrix} G_{cu}(s) \\ G_{uu}(s) \end{bmatrix} U(s) + \begin{bmatrix} G_{cr}(s) \\ G_{ur}(s) \end{bmatrix} Z_r(s), \quad (25)$$

con

$$\begin{aligned} G_{cu}(s) &= \frac{-(m_u f_c s^4 + k_p f_c s^2)}{\alpha_1 s^4 + \alpha_2 s^3 + \alpha_3 s^2 + \alpha_4 s + \alpha_5}, \\ G_{uu}(s) &= \frac{m_c f_c s^4}{\alpha_1 s^4 + \alpha_2 s^3 + \alpha_3 s^2 + \alpha_4 s + \alpha_5}, \\ G_{cr}(s) &= \frac{c_0 k_p s^3 + (k_c + k_0) k_p s^2}{\alpha_1 s^4 + \alpha_2 s^3 + \alpha_3 s^2 + \alpha_4 s + \alpha_5}, \\ G_{ur}(s) &= \frac{(m_c s^2 + c_0 s + k_c + k_0) k_p s^2}{\alpha_1 s^4 + \alpha_2 s^3 + \alpha_3 s^2 + \alpha_4 s + \alpha_5}, \\ \alpha_1 &= m_c m_u, \\ \alpha_2 &= c_0 (m_c + m_u), \\ \alpha_3 &= (m_c + m_u)(k_c + k_0) + m_c k_p, \\ \alpha_4 &= c_0 k_p, \\ \alpha_5 &= k_p (k_c + k_0). \end{aligned} \quad (26)$$

Los parámetros utilizados son los mismos definidos en Morato et al. (2021) y pueden ser vistos en la Tabla 2.

Tabla 1: Parámetros utilizados en el modelo de suspensión del vehículo

Descripción	Valor	Unidad
$m_c$ Masa equiv. acoplamiento chasis	2.27	kg
$m_u$ Masa equiv. acoplamiento rueda	0.32	kg
$k_c$ Coef. muelle de la suspensión	1396	N/m
$k_p$ Coef. muelle que modela la rueda	12270	N/m
$k_0$ Coef. muelle do amortiguador pasivo	170.4	N/m
$c_0$ Coef. rozamiento viscoso pasivo	68.83	N.s/m
$f_c$ Fuerza dinámica del fluido ER	12.59	N

La simulación se realizó en el entorno MATLAB. Los coeficientes de las respuestas al escalón, utilizados en el DMC, se obtuvieron utilizando un período de muestreo de 5 ms y el horizonte del modelo se definió como  $N_{ss} = 100$  muestras. Las respuestas, al escalón en la entrada  $u(t)$ , de las aceleraciones  $A_c(t)$  y  $A_u(t)$  se pueden ver en las Figuras 2(a) y 2(c), respectivamente. Las respuestas, al escalón en el perfil de la carretera  $z_r(t)$ , de las aceleraciones  $A_c(t)$  y  $A_u(t)$  se muestran en las Figuras 2(b) y 2(d), respectivamente.

El algoritmo 1 se implementó en Matlab y el bucle GPAD se tradujo al lenguaje C a través de la biblioteca *codegen*. Los horizontes se ajustaron considerando la dinámica del proceso en bucle abierto, que no tiene retardo, con  $N_1^{(1)} = N_1^{(2)} = 1$ ,  $N_2^{(1)} = 15$ ,  $N_2^{(2)} = 10$  y  $N_u = 5$ . Los pesos de la función objetivo se ajustaron experimentalmente con  $\lambda^{(1)} = 0,03$ ,  $\delta^{(1)} = 0,95$  y  $\delta^{(2)} = 0,075$ . La relación entre los pesos en la aceleración del chasis y la rueda se definió utilizando el cálculo sugerido en Morato et al. (2019), con el fin de priorizar la comodidad de los ocupantes del vehículo. Las restricciones a la acción de control se definieron como  $0 \leq \mathbf{u}(k+j) \leq 1$  con  $j = 0, \dots, N_u - 1$ , por las características del actuador. Las dos tolerancias residuales del algoritmo se ajustaron a  $\epsilon_o = \epsilon_f = 10^{-3}$ . Estos valores se consideraron adecuados para una aplicación práctica del caso de estudio. El problema de optimización a ser resuelto en cada muestra tiene una matriz  $\mathbf{H} \in \mathbb{R}^{5 \times 5}$  y 10 restricciones. Las simulaciones se realizaron en un ordenador con un procesador Intel Core i3 de doble núcleo de 2,40 GHz y 12 GB de memoria RAM. El tiempo de cálculo se midió utilizando la función *tic-toc* disponible en MATLAB.

Se configuró un escenario de simulación con una duración de 15 s con la aplicación de tres perturbaciones en secuencia. El perfil de las perturbación se puede ver en la Figura 3.

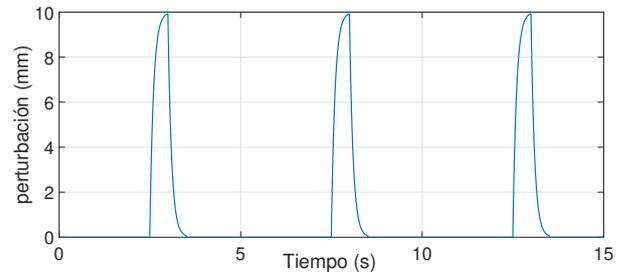
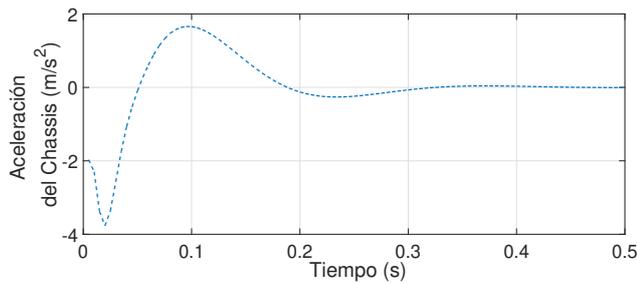


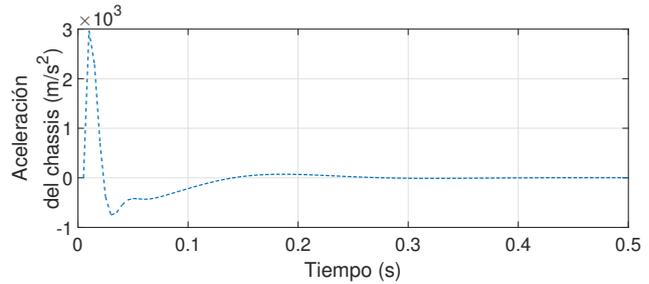
Figura 3: Perfil de la carretera

Los resultados de la simulación para el DMC se pueden ver en la Figura 4. En la Figura 4(a) y en la Figura 4(b), se presentan el perfil de aceleración del chasis y el perfil de aceleración de la rueda, respectivamente. En la Figura 4(c), se muestra la acción de control generada por el DMC. Analizando la respuesta del sistema en bucle cerrado, es posible verificar el rechazo de las perturbaciones causadas por la variación en el perfil de la carretera. La comparación con el sistema con suspensión pasiva<sup>2</sup> se realizó mediante el uso de un índice que considera la raíz cuadrada del error cuadrático medio para el seguimiento de referencias (RMSE, en inglés *Root Mean Squared Error*). Los

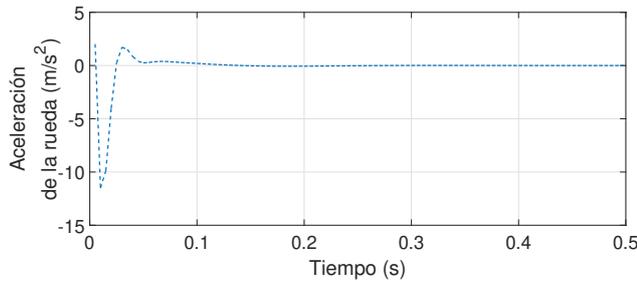
<sup>2</sup>El sistema pasivo corresponde con el mismo modelo del sistema considerando la acción de control nula.



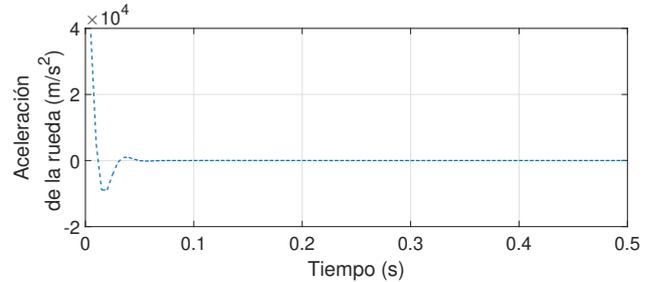
(a) Respuesta de la aceleración del chasis al escalón unitario en la acción de control



(b) Respuesta de la aceleración del chasis al escalón unitario en el perfil de la carretera



(c) Respuesta de la aceleración de la rueda al escalón unitario en la acción de control



(d) Respuesta de la aceleración de la rueda al escalón unitario en el perfil de la carretera

Figura 2: Respuesta al escalón de las aceleraciones para escalones unitarios en  $u(t)$  y en el perfil de la carretera  $z_r(t)$ . Todas las aceleraciones están en  $m/s^2$ .

datos para la comparación de las prestaciones se presentan en la Tabla 2. A partir de los resultados, es posible verificar que el valor del RMSE para la aceleración del chasis con el DMCGPAD es 1,85 veces menor que el valor del RMSE obtenido con la suspensión pasiva. En cuanto a la aceleración de las ruedas, el RMSE para ambos casos fue similar, siendo el RMSE correspondiente al DMCGPAD un 5,4 % más bajo que el RMSE de la suspensión pasiva. Además, los picos de aceleración se reducen en aproximadamente 50 %. De esta forma, los resultados son satisfactorios, dado que la aceleración del chasis es la variable mas importante porque tiene un impacto más directo en la comodidad de los pasajeros. Recordemos que por este motivo, el ajuste del controlador prioriza esta aceleración en la función de coste.

Tabla 2: RMSE de las salidas en el experimento realizado

Método	RMSE - $A_c(t)$ ( $m/s^2$ )	RMSE - $A_u(t)$ ( $m/s^2$ )
DMCGPAD	0.26	2.77
Sistema Pasivo	0.48	2.92

Para el análisis del tiempo de cálculo, se utilizó el comando tic-toc de MATLAB en el bucle de control automático del algoritmo. A modo de comparación, también se implementó el DMC con los optimizadores comerciales Gurobi y Quadprog, ambos con el método de punto interior. El comportamiento en el tiempo del sistema en bucle cerrado es equivalente para todos los optimizadores y, por lo tanto, no se muestra. El perfil de tiempo de cálculo para el DMCGPAD, el DMC con Quadprog y el DMC con Gurobi se puede ver en la Figura 6.

El perfil del número de iteraciones de los optimizadores se puede ver en la Figura 6(a). Se puede observar que el número de iteraciones del DMCGPAD alcanza su punto máximo en los momentos en que existen restricciones activas y presenta

valores mucho más bajos en los demás casos. Esta es una característica esperada para los métodos de primer orden (Ferreau et al., 2017).

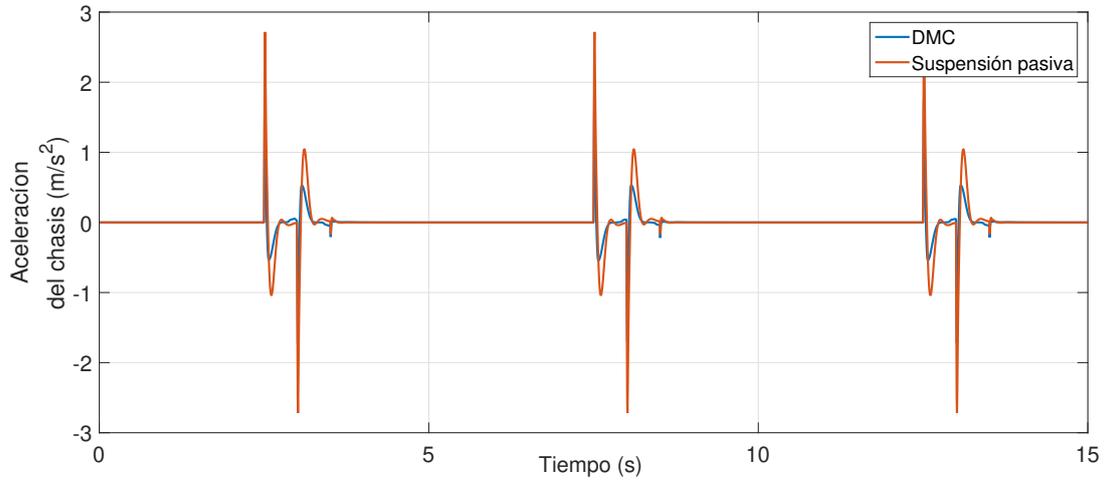
En la Tabla 3 se muestran los tiempos de cálculo (medio y máximo) de la acción de control que se obtuvieron en las simulaciones.

Tabla 3: Tiempos de cálculo de la acción de control

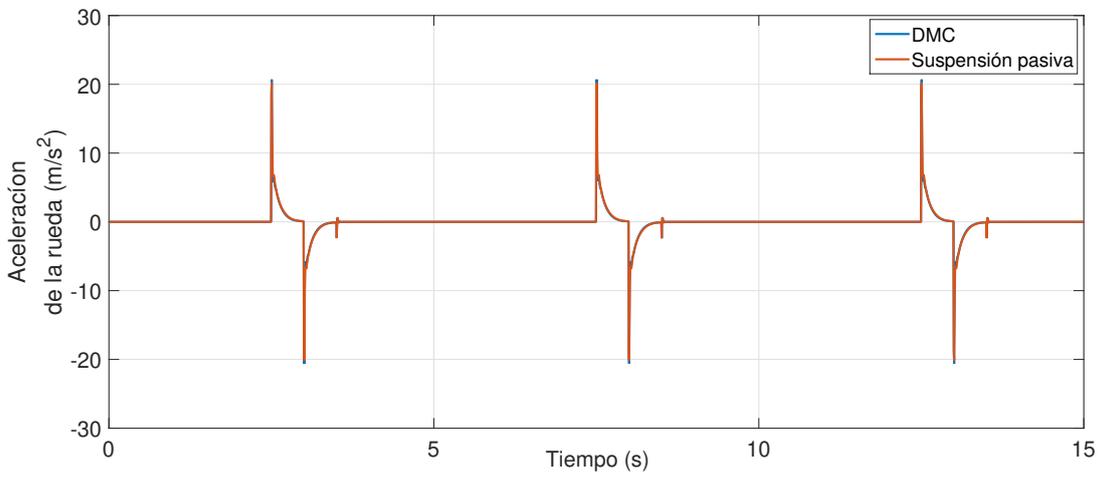
Método	Tiempo máximo	Tiempo medio
DMCGPAD	0,56 ms	0,33 ms
DMCQuadprog	18,15 ms	10,91 ms
DMCGurobi	78,31 ms	15,15 ms

Se puede ver que el tiempo de cálculo del DMCGPAD es menor que el del DMC con el optimizador Quadprog y con el Gurobi. La diferencia es mayor que un orden de magnitud. El tiempo máximo del DMCGPAD de 0,56 ms indica que, incluso en la plataforma MATLAB, que no es optimizada para cálculos rápidos en tiempo real, el algoritmo podría cumplir con el período de muestreo definido en 5 ms.

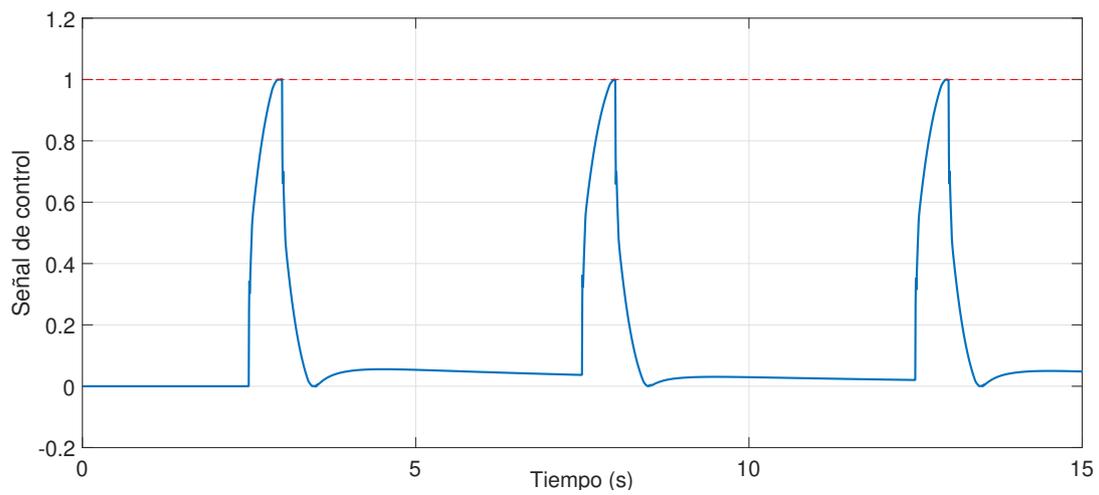
Cabe señalar que al usar un período de muestreo menor se pueden obtener mejores resultados del sistema controlado, especialmente para perturbaciones rápidas o en un escenario donde el vehículo se mueve a alta velocidad. Para investigar el efecto del muestro en las prestaciones del vehículo, se realizó una simulación comparando el comportamiento en bucle cerrado del sistema con dos períodos de muestreo diferentes  $T_s = 5$  ms y  $T_s = 10$  ms. Las ventanas de tiempo asociadas a los horizontes de predicción se mantuvieron iguales en los dos casos, por lo que los valores de  $N_1$ ,  $N_2$  y  $N_u$  se ajustaron según cada período de muestreo. El resultado obtenido en bucle cerrado con los dos controladores, para la aceleración del chasis y la señal de control, se puede ver en la Figura 7. Se observa, en la



(a) Respuesta de la aceleración del chasis



(b) Respuesta de la aceleración de la rueda



(c) Señal de control

Figura 4: Respuesta de las aceleraciones en bucle cerrado para la perturbación causada por la variación en el perfil de la carretera  $z_r(t)$ .

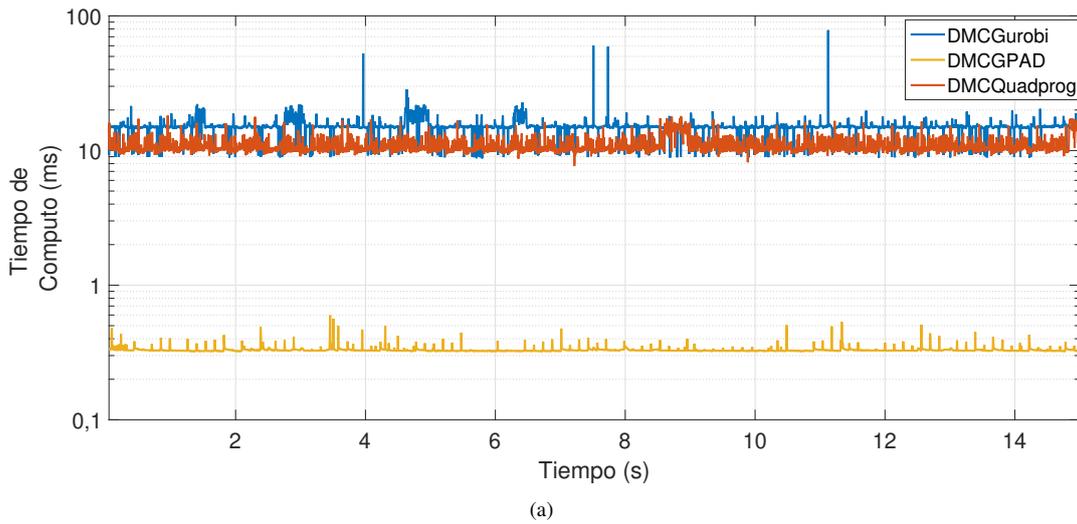


Figura 5: Tiempo de cálculo de la parte en línea del algoritmo de control DMCGPAD, del DMCQuadprog y del DMCGurobi.

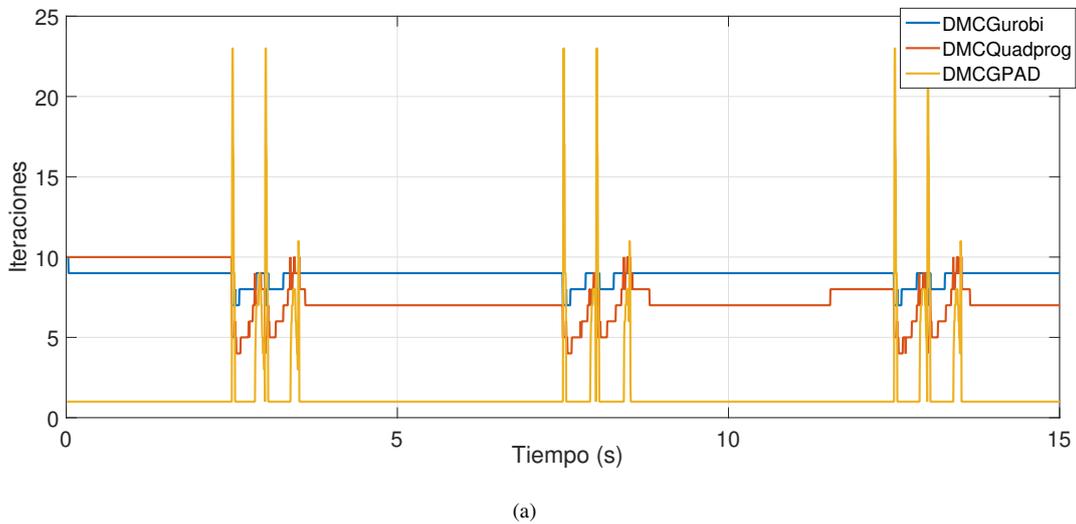
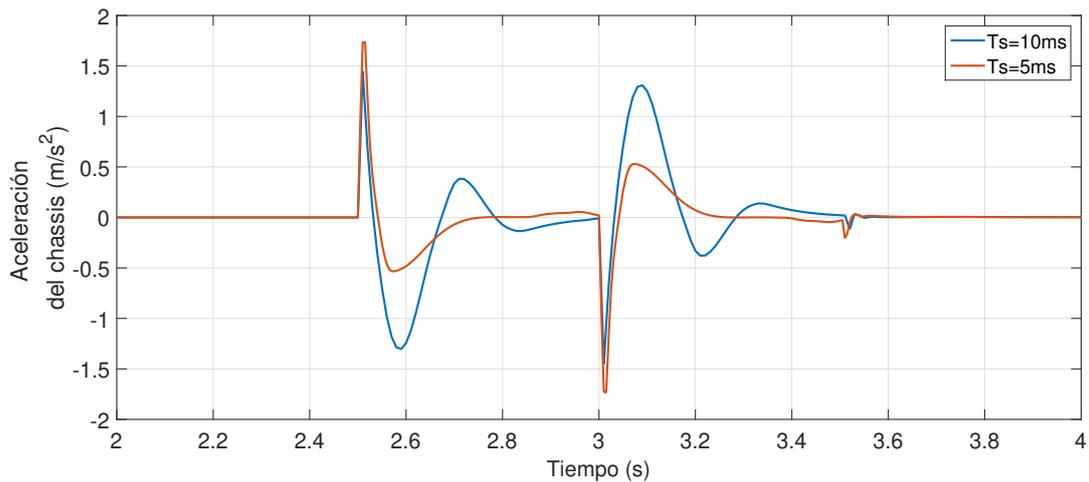


Figura 6: Número de iteraciones de los algoritmos de optimización.

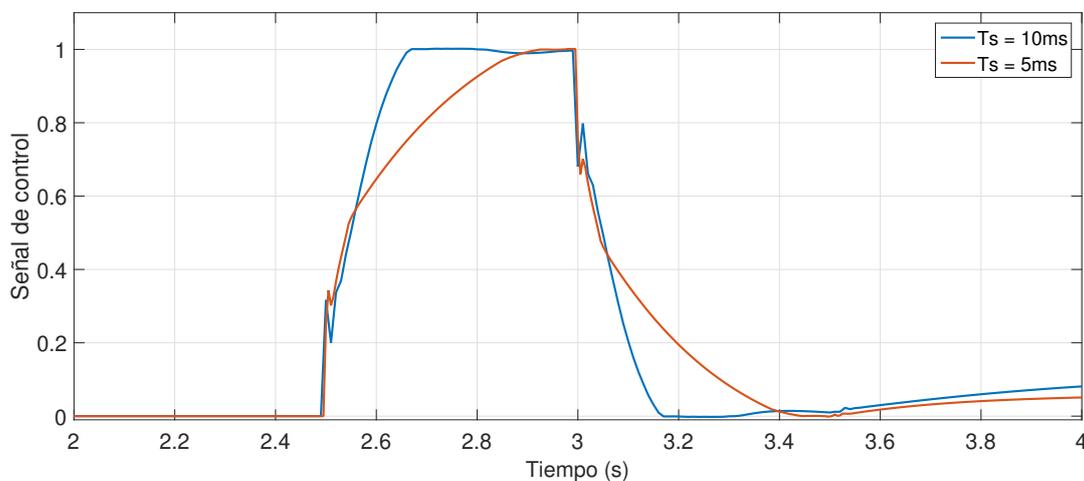
Figura 7(a), que el controlador con el período de muestreo menor, de 5 ms, tiene mejores prestaciones que el controlador con el período de muestreo de 10 ms, rechazando mejor la perturbación. Esta mejora del control con menor  $T_s$  es esperada, ya que las señales de aceleración reaccionan muy rápidamente a la perturbación (en particular la aceleración de la rueda es afectada instantáneamente) y con un menor período de muestreo se consigue una reacción más rápida del control. En el caso de estudio considerado, solo el DMCGPAD podría obtener la acción de control con este período de muestreo, lo que refuerza el uso del algoritmo propuesto.

También se realizó un estudio para evaluar la sensibilidad del algoritmo en relación a diferentes valores en las tolerancias residuales. El resultado de la simulación se puede ver en la Figura 8(a). Los valores máximos de iteraciones en el ciclo GPAD se muestran con tolerancias que van desde  $10^{-2}$  hasta  $10^{-9}$ . Se aprecia un aumento considerable en el número de iteraciones con una menor tolerancia, siendo 11 iteraciones

para el caso  $10^{-2}$  y 146 iteraciones para el caso  $10^{-9}$ . A pesar de este incremento, como el tiempo por iteración del método es bastante bajo, el tiempo máximo observado para el cálculo de la acción de control fue de 0,56 ms en la tolerancia de  $10^{-2}$  a 0,74 ms en la tolerancia de  $10^{-9}$ . Es bien sabido en la literatura que los métodos de primer orden son más adecuados para soluciones de baja y media precisión. Sin embargo, para usar en sistemas de control, las soluciones subóptimas son adecuadas debido a que solo se usa la primera acción de control y una ventana deslizante (Wang and Boyd, 2010). Se puede ver en la Figura 8(b), que el valor del RMSE para la aceleración del chasis es muy poco afectado por las variaciones de la tolerancia y a partir del valor  $10^{-4}$  los cambios son prácticamente imperceptibles. En los sistemas reales existen todavía otras limitaciones prácticas, como la resolución de sensores y actuadores, por ejemplo. Un estudio de la sensibilidad del GPAD en relación al aumento del horizonte de control en el contexto de GPC se puede ver en Peccin et al. (2021).



(a) Respuesta de la aceleración del chasis con diferentes períodos de muestreo



(b) Señal de control con diferentes períodos de muestreo

Figura 7: Respuesta de la aceleración en bucle cerrado para la perturbación causada por la variación en el perfil de la carretera con diferentes períodos de muestreo.

## 5. Implementación en FPGA

El algoritmo DMCGPAD se incorporó a un FPGA y se evaluó el tiempo de cálculo. Esta implementación fue posible ya que el algoritmo propuesto solo usa recursos computacionales y operaciones modestos. Se utilizó una FPGA de Altera, familia MAX 10, modelo 10M50DAF484C7G. La familia MAX 10 contiene un convertidor analógico/digital y 50 000 elementos lógicos programables. El algoritmo 1 se implementó con un lenguaje de descripción de hardware llamado Verilog. Se utilizó una aritmética de punto fijo de 16 bits para representar los números, con 10 bits para los decimales. Se usó una máquina de estado para sincronizar el algoritmo y todas las operaciones matriciales se implementaron en computación paralela.

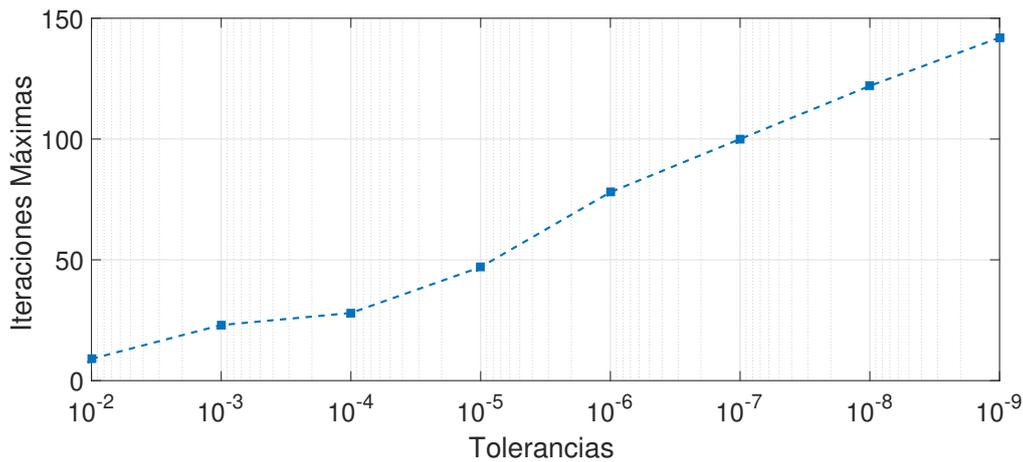
Los parámetros de configuración del controlador se pasan a través de una comunicación externa (USB-Blaster) con el procesador NIOS II implementado en una FPGA. La comunicación entre los bloques lógicos del DMC y el código incrustado del NIOS II se realizó a través del bus Avalon. El bloque DMC crea una instancia de los bloques GPAD y Free Response. Las bibliotecas de multiplicación de matrices y las operaciones básicas en aritmética de punto fijo también se implementaron directamente en la lógica personalizada. El algoritmo de multiplica-

ción de matrices paralelas tiene una propiedad de complejidad dada por  $O(\lceil \log_{10}(n) \rceil + 1)$  (Peccin et al., 2020). Los valores de tiempo de ejecución se obtuvieron utilizando un osciloscopio y conmutación de puerta digital. Se pueden ver más detalles sobre la arquitectura de implementación de FPGA en Peccin et al. (2020).

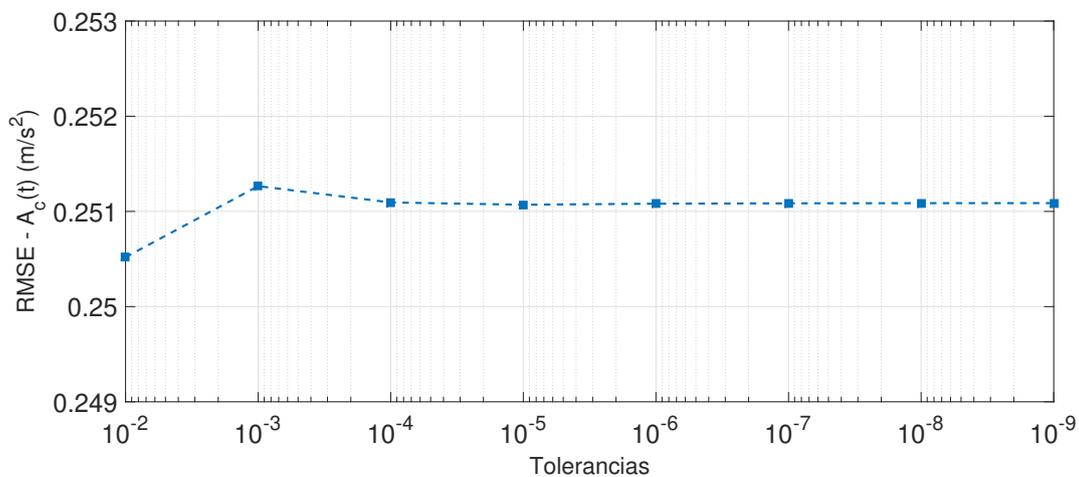
El peor caso observado en la simulación para el estudio de caso en cuestión, con 23 iteraciones del GPAD, se replicó para la implementación en el FPGA. Los resultados obtenidos, con lo tiempo de ejecución en el peor de los casos (WCET, en inglés *Worst Case Execution Time*) en las funciones principales, se presentan en la Tabla ???. El tiempo de ejecución es rápido, con un valor máximo de  $7,64 \mu\text{s}$ , lo que permite el uso de DMC en procesos con dinámica rápida utilizando periodos de muestreo inferiores a 1 ms.

## 6. Conclusiones

Este trabajo propuso un algoritmo DMC basado en el método de gradiente proyectado acelerado dual, llamado DMCGPAD. El algoritmo DMCGPAD presentó un tiempo de ejecución del orden de ms, considerando la optimización en línea,



(a) Número de iteraciones del GPAD con diferentes tolerancias.



(b) RMSE para diferentes tolerancias del GPAD

Figura 8: Análisis del número máximo de iteraciones y del RMSE para diferentes tolerancias del GPAD

siendo adecuado para el caso de estudio en cuestión, que utiliza un tiempo de muestreo de 5 ms. Las ventajas del DMCGPAD propuesto son muy importantes en casos donde el problema de optimización tiene dimensiones que exigen mucho espacio de memoria para el almacenamiento de datos. La implementación del algoritmo de control en una FPGA Max 10 demostró ser muy rápida, con un tiempo de cálculo de  $7,64 \mu\text{s}$  para el caso de estudio utilizado en este trabajo. Estos resultados contribuyen a facilitar el uso del DMC en los niveles más bajos de automatización y así poder aprovechar sus ventajas inherentes en sistemas que se operan típicamente con controladores clásicos, debido a la existencia de dinámicas rápidas.

### Agradecimientos

Los autores agradecen el apoyo financiero dado por CNPq – proyectos 304032/2019-0 y 315546/2021-2.

### Referencias

Ahmadi-Moshkenani, P., Johansen, T. A., Olaru, S., 2018. Combinatorial approach toward multiparametric quadratic programming based on characterizing adjacent critical regions 63 (10), 3221–3231.

- Borrelli, F., Baotić, M., Pekar, J., Stewart, G., 2010. On the computation of linear model predictive control laws. *Automatica* 46 (6), 1035 – 1041.
- Cairano, S. D., Brand, M., Bortoff, S. A., 2013. Projection-free parallel quadratic programming for linear model predictive control. *Int. J. Control* 86 (8), 1367–1385.
- Camacho, E., Bordons, C., 2004. *Model Predictive Control*. Advanced Textbooks in Control and Signal Processing. Springer, London.
- Cimini, G., Bemporad, A., Dec 2017. Exact complexity certification of active-set methods for quadratic programming. *IEEE Trans. Automat. Contr.* 62 (12), 6094–6109.
- Cutler, C. R., Ramaker, B. L., 1980. Dynamic matrix control: A computer control algorithm. In: *Proc. Automatic Control Conference*. Vol. 17. p. 72.
- Fernandes, D., Haque, M. E., Palanki, S., Rios, S. G., Chen, D., 2020. DMC controller design for an integrated allam cycle and air separation plant. *Comput. Chem. Eng.* 141, 107019.
- Ferreau, H., Almér, S., Verschueren, R., Diehl, M., Frick, D., Domahidi, A., Jerez, J., Stathopoulos, G., Jones, C., Dec 2017. Embedded optimization methods for industrial automatic control. In: *Proc. 20th IFAC World Congr.* Toulouse, France.
- Goldstein, T., O’Donoghue, B., Setzer, S., Baraniuk, R., 2014. Fast alternating direction optimization methods. *SIAM J. on Imaging Sciences* 7 (3), 1588–1623.
- He, X., Lima, F. V., 2019. Development and implementation of advanced control strategies for power plant cycling with carbon capture. *Comput. Chem. Eng.* 121, 497 – 509.
- Herceg, M., Jones, C. N., Morari, M., 2015. Dominant speed factors of active set methods for fast MPC. *Optim. Contr. Appl. Met.* 36 (5), 608–627.
- Kiencke, U., Nielsen, L., 2000. *Automotive Control Systems: For Engine, Dri-*

- veline and Vehicle, 1st Edition. Springer-Verlag, Berlin, Heidelberg.
- Kvasnica, M., Takács, B., Holaza, J., Di Cairano, S., 2015. On region-free explicit model predictive control. In: Proc. 54th IEEE Conf. on Decision and Control (CDC). pp. 3669–3674.
- Lee, J. H., Morari, M., Garcia, C. E., 1994. State-space interpretation of model predictive control. *Automatica* 30 (4), 707–717.
- Lima, D. M., Normey-Rico, J. E., Plucenio, A., Santos, T. L. M., Gomes, M. V., 2014. Improving robustness and disturbance rejection performance with industrial MPC. In: Proc. 20th Brazilian Conference on Automation (CBA). pp. 3229–3236.
- Morato, M. M., Normey-Rico, J. E., Sename, O., 2021. An input-to-state stable model predictive control framework for Lipschitz nonlinear parameter varying systems. *International Journal of Robust and Nonlinear Control* 31 (17), 8239–8272.
- Morato, M. M., Q., N. M., Sename, O., Dugard, L., 2019. Design of a fast real-time LPV model predictive control system for semi-active suspension control of a full vehicle. *Journal of the Franklin Institute* 356 (3), 1196–1224.
- Nesterov, Y., 1983. A method of solving a convex programming problem with convergence rate  $o(1/k^2)$ . *Soviet Mathematics Doklady* 27 (2), 372–376.
- O'Donoghue, B., Stathopoulos, G., Boyd, S., Nov 2013. A splitting method for optimal control 21 (6), 2432–2442.
- Patrinos, P., Bemporad, A., Jan 2014. An accelerated dual gradient-projection algorithm for embedded linear model predictive control 59 (1), 18–33.
- Peccin, V. B., Lima, D. M., Flesch, R. C. C., Normey-Rico, J. E., 2019. Fast generalized predictive control based on accelerated dual gradient projection method. In: Proc. 12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems (DYCOPS). pp. 474–479.
- Peccin, V. B., Lima, D. M., Flesch, R. C. C., Normey-Rico, J. E., 2020. Fast constrained generalized predictive control with ADMM embedded in an FPGA. *IEEE Latin America Trans.* 18 (2), 422–429.
- Peccin, V. B., Lima, D. M., Flesch, R. C. C., Normey-Rico, J. E., 2021. Fast algorithms for constrained generalised predictive control with on-line optimisation. *IET Control Theory & Applications* 15 (4), 545–558.
- Pistikopoulos, E. N., Diangelakis, N. A., Oberdieck, R., Papanthanasios, M. M., Nascu, I., Sun, M., 2015. PAROC—An integrated framework and software platform for the optimisation and advanced model-based control of process systems. *Chem. Eng. Science* 136, 115–138.
- Pu, Y., Zeilinger, M. N., Jones, C. N., Feb 2017. Complexity certification of the fast alternating minimization algorithm for linear MPC 62 (2), 888–893.
- Roldao-Lopes, A., Shahzad, A., Constantinides, G. A., Kerrigan, E. C., April 2009. More flops or more precision? Accuracy parameterizable linear equation solvers for model predictive control. In: Proc. 17th IEEE Symposium on Field-Programmable Custom Computing Machines. pp. 209–216.
- Wang, J., Xu, Z., Song, C., Yao, Y., Zhao, J., 2020. A distributed model predictive control algorithm with the gap metric output feedback decoupling. *Comput. Chem. Eng.*, 107167.
- Wang, Y., Boyd, S., March 2010. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology* 18 (2), 267–278.
- Wills, A., Mills, A., Ninness, B., 2011. FPGA implementation of an interior-point solution for linear model predictive control. In: Proc. 18th IFAC World Congress.
- Wojtulewicz, A., Ławryńczuk, M., 2018. Implementation of multiple-input multiple-output dynamic matrix control algorithm for fast processes using field programmable gate array. In: Proc. 15th IFAC Conference on Programmable Devices and Embedded Systems (PDeS). pp. 324–329.