Alternating Direction of Multipliers Method for Block Circulant Model Predictive Control

Idris Kempf*, Paul J. Goulart and Stephen Duncan

Abstract— **This paper deals with model predictive control problems for large-scale dynamical systems with cyclic symmetry. Based on the properties of block circulant matrices, we use the discrete Fourier transformation to block diagonalize and truncate the original finite-horizon optimal control problem. Using this coordinate transformation, we develop a modified alternating direction of multipliers method (ADMM) algorithm for general constrained quadratic programs with block circulant blocks. We test our modified algorithm using random data and in a traffic flow control example and show that the coordinate transformation significantly increases the computation speed.**

Index Terms— **Model Predictive Control (MPC), Alternating Direction of Multipliers Method (ADMM), Block Circulant Systems, Quadratic Program, Traffic Flow Control**

I. INTRODUCTION

The advantages of model predictive control (MPC) for constraint handling and feedforward disturbance modelling are widely recognised. However, its applicability is limited by the requirement to solve optimization problems in real-time to compute the control law. This constraint has inhibited the application of MPC to large-scale and high speed applications. While some approaches for accelerating the computing speed have focused on implementing optimization routines on specialised high-performance hardware [1], other approaches have exploited the particular symmetric structure encountered in some classes of large-scale problems [2].

Block circulant (BC) matrices are a subclass of Toeplitz matrices and have been extensively studied in the past [3]. These matrices are known to be amenable to decompositions that are kept throughout optimal control problems [4], [5]. The projection into the Fourier domain leads to an advantageous structure that can be exploited to significantly increase the computation speed of matrix operations.

In this paper we address systems with cyclic symmetry resulting in a BC structure. These systems can be interpreted as the symmetric interconnection of many subsystems, where each subsystem interacts in an identical way with its neighbors [4]. Circulant systems can be found in a variety of applications, including traffic flow control [6], cross-directional control [7], particle accelerator control [8] and in the approximation of partial differential equations [9]. The mathematical properties of these systems have already been exploited in controller design [5], [10], stability analysis [11] and subspace identification [12]. We investigate the properties that a constrained quadratic program (CQP) inherits from a block circulant MPC (BCMPC) problem. The main results of the paper show how exploiting the properties of the resulting CQP, such as projecting the problem into Fourier space and truncating the vector of decision variables, can reduce the computational cost when it is solved using the alternating direction of multipliers method [13].

This paper is structured as follows. In Section II, the linear MPC problem and the alternating direction of multipliers method (ADMM) – an algorithm which is particularly suitable for solving the latter optimization problem – are introduced. Since we are concerned with the analysis of systems with BC symmetry, we introduce the notion of BC matrices in Section III. Furthermore, the BCMPC problem is formally defined and necessary conditions for its decomposition are stated. In Section IV, we define a CQP with BC blocks and show how MPC problems with BC data can be written in this form. The BC decomposition is then applied to the CQP and a modified ADMM algorithm is then formulated for this problem. In Section V, we compare the performance of the original and modified ADMM algorithms. For the sake of comparison, both algorithms have been implemented in Matlab and tested on two illustrative examples.

Notation and Definitions Let $\otimes$ denote the Kronecker product and $\oplus$ denote the direct sum (i.e. the block diagonal concatenation) of two matrices. Let $\mathbb{I}_n$ represent the identity matrix in $\mathbb{R}^{n\times n}$. For a scalar, vector or matrix a, let $\bar{a}$ denote its complex conjugate; Let $Re(a)$ and $Im(a)$ denote its real and imaginary part, respectively; Let a^H denote its Hermitian transpose. Let $\text{diag}\{a_1,\dots,a_n\}$ denote a diagonal matrix with diagonal elements $a_1,\dots,a_n$.

II. PROBLEM STATEMENT

A. Model Predictive Control

Given a discrete-time linear dynamical system and an initial condition $x(t)$ at time t, a standard MPC scheme computes a control law by predicting the future evolution of the system and minimizing a quadratic objective function over some planning horizon T. This can be achieved via repeated solution of the following quadratic program (QP):

$$\min \sum_{k=0}^{T-1} x_k^\mathrm{T} Q x_k + u_k^\mathrm{T} R u_k + x_N^\mathrm{T} P x_N \tag{1a}$$

$$s.t.\ \ x_{k+1} = Ax_k + Bu_k \qquad x_0 = x(t) \tag{1b}$$

$$y_k = Cx_k + Du_k \tag{1c}$$

$$\underline{y} \le y_k \le \bar{y}, \tag{1d}$$

[*] Corresponding author: idris.kempf@eng.ox.ac.uk. All authors are with the Department of Engineering Science, University of Oxford, Oxford, UK. This research is supported by the Engineering and Physical Sciences Research Council (EPSRC) with a Diamond CASE studentship.

for $k = 0, \dots, T-1$, returning at each step the optimal first input stage u_0^* as a control law. The constraints on the states $x_k \in \mathbb{R}^{n_x}$ and the inputs $u_k \in \mathbb{R}^{n_u}$ are lumped into the variable $y_k \in \mathbb{R}^{n_y}$. The stability of the state is guaranteed if $P = P^{\mathrm{T}} \succ 0$ is obtained from the discrete-time algebraic Riccati equation (DARE) associated to the unconstrained infinite horizon regulator problem. The QP (1) has a unique solution if $R \succ 0$, $Q \succeq 0$ and the pairs (A, B) and $(A, Q^{\frac{1}{2}})$ are controllable and observable, respectively [14, Chapter 12]. Throughout the paper, we will assume that (1) admits a unique solution and is recursively feasible.

By eliminating the state variables $(x_1, \dots, x_N)$ and defining $z := (u_0, \dots, u_{T-1})^{\mathrm{T}}$ and $v := (y_0, \dots, y_{T-1})^{\mathrm{T}}$, (1) can be reformulated as

$$\min \ \frac{1}{2} z^{\mathrm{T}} J z + q^{\mathrm{T}} z \tag{2a}$$

$$s.t. \ Kz - v = 0 \tag{2b}$$

$$\underline{v} \le v \le \bar{v}. \tag{2c}$$

Note that $(\bar{v}, \underline{v}, q)$ depend on x_0. The matrices (J, K) and vectors $(\underline{v}, \bar{v}, q)$ in (2) are defined as

$$J := G^{\mathrm{T}} \left((\mathbb{I}_T \otimes Q) \oplus P \right) G + (\mathbb{I}_T \otimes R), \tag{3a}$$

$$K := [\mathbb{I}_T \otimes C \,|\, 0]\, G + (\mathbb{I}_T \otimes D), \tag{3b}$$

$$\underline{v} := (\mathbf{1}_T \otimes \underline{y}) - [\mathbb{I}_T \otimes C \,|\, 0]\, H x_0, \tag{3c}$$

$$\bar{v} := (\mathbf{1}_T \otimes \bar{y}) - [\mathbb{I}_T \otimes C \,|\, 0]\, H x_0, \tag{3d}$$

$$q := G^{\mathrm{T}} H x_0, \tag{3e}$$

where $\mathbf{1}_T$ is a vector of ones of length T and G and H arise from elimination of the equality constraints in (1b), i.e. from setting $X = (x_0, \dots, x_T)^{\mathrm{T}}$ and writing (1b) as $X = Gz + Hx_0$. Note that $J \succ 0$ because $R \succ 0$ by assumption.

B. ADMM Algorithm

We consider application of the ADMM to the solution of (2), and will follow the specific ADMM formulation presented in [13] throughout. The method is summarized in Algorithm 1. The augmented Lagrangian for (2) can be written as

$$\begin{aligned} L(z, v, \gamma) = & \frac{1}{2} z^{\mathrm{T}} J z + q^{\mathrm{T}} z + \frac{\rho}{2} \|Kz - v\|_2^2 \\ & + \gamma^{\mathrm{T}} (Kz - v) + \mathcal{I}_{[\underline{v}, \bar{v}]}(v), \end{aligned} \tag{4}$$

where $\mathcal{I}_{[\underline{v}, \bar{v}]}$ is the indicator function for the set $\{v \mid \underline{v} \le v \le \bar{v}\}$ and the penalty parameter $\rho > 0$ and the dual variables γ are associated with the constraint (2b). ADMM solves (2) by repeatedly minimizing (4) w.r.t. z and v and updating the dual variables γ using an approximate gradient ascent method. Even though the assumptions in section II-A guarantee the convergence of Algorithm 1, it is common practice to limit it to a maximum number of iterations i_{max}.

Algorithm 1 ADMM for MPC

```
Input: State $x(t)$
Output: Input $u(t)$
Set $x_0 = x(t)$ and $v^0, \gamma^0 = 0$; compute $\underline{v}, \bar{v}$ and $q$
for $i = 1$ to $i_{max}$ do
    Update $z^i$ using (SP1)
    Update $v^i$ using (SP2)
    Update $\gamma^i$ using (SP3)
    if $\|v^i - v^{i-1}\|_2^2 < \epsilon$ and $\|\gamma^i - \gamma^{i-1}\|_2^2 < \epsilon$ then
        break
    end if
end for
return $u(t) = (z_1, \dots, z_{n_u})^{\mathrm{T}}$
```

After initialization[1], Algorithm 1 first minimizes (4) w.r.t. to z, which, after completing the square, is equivalent to

$$z^i = \arg\min_{z} \frac{1}{2} z^{\mathrm{T}} J z + q^{\mathrm{T}} z + \frac{\rho}{2} \|Kz - v^{i-1} + \rho^{-1} \gamma^{i-1}\|_2^2, \tag{5}$$

with iteration index $i = 1, \dots, i_{max}$. Since (5) is an unconstrained QP, its derivative can be set to zero and the resulting linear system can then be solved from

$$\left(J + \rho K^{\mathrm{T}} K \right) z^i = K^{\mathrm{T}} (\rho v^{i-1} - \gamma^{i-1}) - q. \tag{SP1}$$

The linear system (SP1) always admits a solution because $J + \rho K^{\mathrm{T}} K \succ 0$ under the assumptions from section II-A.

With z^i obtained, Algorithm 1 then minimizes (4) w.r.t. v by solving

$$v^i = \arg\min_{\underline{v} \le v \le \bar{v}} \|Kz^i - v + \rho^{-1} \gamma^{i-1}\|_2^2. \tag{6}$$

The solution to (6) can be written as

$$v^i = \operatorname*{sat}_{[\underline{v}, \bar{v}]} \left\{ Kz^i + \rho^{-1} \gamma^{i-1} \right\}, \tag{SP2}$$

where the saturation function limits its argument to $\underline{v}$ and $\bar{v}$.

Finally, algorithm 1 updates the dual variable γ according to

$$\gamma^i = \gamma^{i-1} + \rho (Kz^i - v^i). \tag{SP3}$$

Subproblems (SP1) - (SP3) are repeated until some convergence criterion or the maximum number of iterations is reached. Proofs and other variants of the ADMM can be found in [13], [15].

III. Block Circulant Systems

A. Preliminaries

Definition 1 (Block Circulant Matrices [3, Chapter 5]): Let $\mathcal{BC}(n, p, m) \subseteq \mathbb{R}^{np \times nm}$ denote the set of real *block circulant matrices* in the form

$$B = \begin{bmatrix} b_0 & b_1 & b_2 & \dots & b_{n-1} \\ b_{n-1} & b_0 & b_1 & \dots & b_{n-2} \\ \vdots & & & & \vdots \\ b_1 & b_2 & b_3 & \dots & b_0 \end{bmatrix}, \tag{7}$$

[1] Note that we assume that ADMM is *cold-started* in Algorithm 1 at each time step for simplicity, but in practice one would warm start the variables (v^0, γ^0) from a previous solution.

where $b_i \in \mathbb{R}^{p \times m}$. If $p = m = 1$, then B is simply called a *circulant matrix*.

The set of BC matrices $\mathcal{BC}(n, p, m)$ has a number of very useful basic properties [3, Chapter 5.6], including closure under matrix addition, multiplication and transposition. Moreover, if $A \in \mathcal{BC}(n, p, m)$ and $B \in \mathcal{BC}(n, m, r)$ then $AB \in \mathcal{BC}(n, p, r)$.

Definition 2 (Fourier Matrix): Let $F_n \in \mathbb{C}^{n \times n}$ with $F_n F_n^H = \mathbb{I}_n$ denote the *Fourier matrix*, defined as

$$F_n = \frac{1}{\sqrt{n}} \begin{bmatrix} w_0 & w_1 & \dots & w_{n-1} \end{bmatrix}, \tag{8}$$

where the vectors $w_j = \begin{pmatrix} 1 & \rho_j & \rho_j^2 & \dots & \rho_j^{n-1} \end{pmatrix}^{\mathrm{T}}$ are mutually orthogonal and $\rho_j = e^{i\frac{2\pi}{n}j}$ are complex roots of unity.

The matrix (8) is called the Fourier matrix since the Fourier coefficients of the discrete Fourier transformation of a vector $x \in \mathbb{R}^n$ can be obtained from the product $F_n x$ (or more efficiently using a fast Fourier transformation [16]).

A fundamental property of BC matrices is that every BC matrix $B \in \mathcal{BC}(n, p, m)$ is diagonalized by the same pair of matrices:

Theorem 1 (Block Diagonalization [12]): For $B \in \mathbb{R}^{np \times nm}$, it holds that $\hat{B} = (F_n \otimes \mathbb{I}_p)^H B (F_n \otimes \mathbb{I}_m)$ is block diagonal iff $B \in \mathcal{BC}(n, p, m)$. The blocks $\nu_j \in \mathbb{C}^{p \times m}$ of $\hat{B} = \operatorname{diag}\{\nu_0, \dots, \nu_{n-1}\}$ are obtained as

$$\nu_j = b_o + b_1 \rho_j + \dots + b_{n-1} \rho_j^{n-1}, \; j = 0, \dots, n-1. \tag{9}$$

From the structure of F_n, the following corollary can be established on the structure of the blocks $\nu_0, \dots, \nu_{n-1}$:

Corollary 1 (Pattern of Complex Conjugates [12]): Define $n_c := \frac{n}{2}$ for even and $n_c := \frac{n-1}{2}$ for odd n, respectively. The blocks ν_j, $j = 0, \dots, n-1$, from (9) have the following pattern of complex conjugates: If n is odd, then ν_0 is real while $(\nu_1, \dots, \nu_{n_c}) = (\bar{\nu}_{n-1}, \dots, \bar{\nu}_{\frac{n+1}{2}})$. If n is even, then ν_0 and ν_{n_c} are real while $(\nu_1, \dots, \nu_{\frac{n}{2}-1}) = (\bar{\nu}_{n-1}, \dots, \bar{\nu}_{\frac{n}{2}+1})$.

Using the shuffle permutation matrix Π_n^m from [16], we can rewrite $F_n \otimes \mathbb{I}_m$ as $F_n \otimes \mathbb{I}_m = (\Pi_n^m)^{\mathrm{T}} (I_m \otimes F_n) \Pi_n^m$. As a consequence, $(F_n \otimes \mathbb{I}_m)x$ with $x \in \mathbb{R}^{nm}$ can be computed by permuting the vector, applying a sequence of m fast Fourier transformations (FFTs) and applying the inverse permutation. This implies that $(F_n \otimes \mathbb{I}_m)x$ is of complexity $\mathcal{O}(mn \log n)$ compared to $\mathcal{O}(m^2 n^2)$ for general matrix-vector multiplication. Note that the pattern of complex conjugates also holds for the n blocks of the vector $(F_n \otimes \mathbb{I}_m)x$.

A consequence of Corollary 1 for some matrix $B \in \mathcal{BC}(n, p, m)$ is that, after block diagonalization, its data is uniquely determined by the first $n_c + 1$ blocks of the block diagonal matrix $\hat{B}$. It is therefore sufficient to examine the first $n_c + 1$ blocks of $\hat{B}$, which will be exploited in the subsequent analysis. For this reason, the truncation and its counterpart operation, the augmentation, are formally defined:

Definition 3 (Truncation and Augmentation): Given $A = \operatorname{diag}\{a_1, \dots, a_n\} \in \mathbb{C}^{np \times nm}$, $x \in \mathbb{C}^{np}$ and n even (odd), let trunc be the operator that extracts the first $\frac{n}{2}$ ($\frac{n-1}{2}$) blocks of A and the first $\frac{n}{2}p$ ($\frac{n-1}{2}p$) rows of x. Conversely, let aug be the inverse operator which accepts a truncated vector ${}_\perp x = (x_1, \dots, x_{\frac{n}{2}})^{\mathrm{T}} \in \mathbb{C}^{\frac{n}{2}p}$ with $x_i \in \mathbb{C}^p$ and returns $x = (x_1, \dots, x_{\frac{n}{2}}, \bar{x}_{\frac{n}{2}-1}, \dots, \bar{x}_2)^{\mathrm{T}} \in \mathbb{C}^{np}$ for even n. For odd n, operator aug accepts a vector ${}_\perp x = (x_1, \dots, x_{\frac{n-1}{2}})^{\mathrm{T}} \in \mathbb{C}^{\frac{n-1}{2}p}$ with $x_i \in \mathbb{C}^p$ and returns $x = (x_1, \dots, x_{\frac{n-1}{2}}, \bar{x}_{\frac{n-1}{2}}, \dots, \bar{x}_2)^{\mathrm{T}} \in \mathbb{C}^{np}$. Define aug in a similar way when invoked with a truncated block diagonal matrix ${}_\perp A = \operatorname{diag}\{a_1, \dots, a_{\frac{n}{2}}\}$. In addition, define $\operatorname{trunc}_N := \mathbb{I}_N \otimes \operatorname{trunc}$ and $\operatorname{aug}_N := \mathbb{I}_N \otimes \operatorname{aug}$, i.e. the operators aug and trunc applied N times.

Note that aug and trunc are linear operators. In addition, it holds that $\operatorname{trunc}\{Jz\} = \operatorname{trunc}\{J\} \operatorname{trunc}\{z\}$ and $\operatorname{aug}\{{}_\perp J \, {}_\perp z\} = \operatorname{aug}\{{}_\perp J\} \operatorname{aug}\{{}_\perp z\}$ for block diagonal matrices $J, {}_\perp J$ and vectors $z, {}_\perp z$ of compatible dimensions.

B. Block Circulant MPC

We are concerned in particular with dynamic linear systems where the matrices present in (1) are BC. More formally, a BCMPC problem is defined as follows:

Definition 4 (Block Circulant MPC): Consider (1) with $x_k \in \mathbb{R}^{nn_x}, u_k \in \mathbb{R}^{nn_u}$ and $y_k \in \mathbb{R}^{pnn_y}$, where p is a positive integer, and partition matrices C and D as $C = [C_1, \dots, C_p]^{\mathrm{T}}$ and $D = [D_1, \dots, D_p]^{\mathrm{T}}$, respectively. We say that (1) is a *block circulant MPC problem* of order n if the following conditions holds:

$$\begin{aligned} A, Q &\in \mathcal{BC}(n, n_x, n_x), & B &\in \mathcal{BC}(n, n_x, n_u), \\ R &\in \mathcal{BC}(n, n_u, n_u), & C_i &\in \mathcal{BC}(n, n_y, n_x), \\ D_i &\in \mathcal{BC}(n, n_y, n_u), \end{aligned}$$

for $i = 1, \dots, p$.

Any model satisfying the above conditions can be interpreted as a periodic interconnection of n identical subsystems with identical constraints and objective function penalties. Introducing the integer p allows for multiple constraint sets, e.g. to restrict the state x_k and the input u_k separately. In addition, note that if a system satisfies the conditions in Definition 4, then the terminal cost matrix P solving the corresponding DARE is BC as well, i.e. $P \in \mathcal{BC}(n, n_x, n_x)$ [17], [18].

IV. Block Circulant ADMM Algorithm

A. Constrained QP with Block Circulant Blocks

Definition 5 (Constrained block circulant QP): The following real valued constrained QP,

$$\min \; \frac{1}{2} z^{\mathrm{T}} J z + q^{\mathrm{T}} z \tag{10a}$$

$$s.t. \; Kz - v = 0 \tag{10b}$$

$$\underline{v} \leq v \leq \bar{v}, \tag{10c}$$

is called a *constrained block circulant QP* of order n if there exists a partitioning of vectors z and v into N_z and N_v segments of lengths $l_z^1, \dots, l_z^{N_z}$ and $l_v^1, \dots, l_v^{N_v}$, respectively, that partition matrices J and K such that all blocks are BC matrices of order n,

$$J_{kj} \in \mathcal{BC}(n, l_z^k, l_z^j), \qquad K_{wj} \in \mathcal{BC}(n, l_v^w, l_z^j),$$

for $k, j = 1, \dots, N_z$ and $w = 1, \dots, N_v$.

Due to the algebraic properties of BC matrices summarized in Section III-A, the BC structure of a BCMPC problem is preserved when the MPC problem is formulated as a QP such as in (2). The following corollary connects a BCMPC problem from Definition 4 to the constrained block circulant quadratic program (CBCQP) from Definition 5:

Corollary 2 ([18]): A block circulant MPC problem leads to a CBCQP with $N_z = T$, $N_v = Tp$ and $J_{kj} \in \mathcal{BC}(n, n_u, n_u)$ and $K_{wj} \in \mathcal{BC}(n, n_y, n_u)$.

As the matrices in Definition 5 are composed of BC blocks, the results from Sections III-A and III-B suggest that there exists a coordinate transformation $(\tilde{z}, \tilde{v}) = (\psi_z^H z, \psi_v^H v)$ that block diagonalizes each block of K and J. However, how the complex-valued transformation affects the minimization in (10) is not immediately obvious. The following theorem answers this question:

Theorem 2 (Decomposition of CBCQP [18]): Given a CBCQP of order n, then the following CBCQP,

$$\min_{\tilde{z} \in \mathcal{S}_z, \tilde{v} \in \mathcal{S}_v} \frac{1}{2} \tilde{z}^H \tilde{J} \tilde{z} + \tilde{q}^H \tilde{z} \tag{11a}$$

$$s.t. \;\; \tilde{K} \tilde{z} - \tilde{v} = 0 \tag{11b}$$

$$\underline{v} \leq \psi_v \tilde{v} \leq \bar{v}, \tag{11c}$$

where $\tilde{q} = \psi_z^H q$, $\tilde{J} = \psi_z^H J \psi_z$, $\tilde{K} = \psi_v^H K \psi_z$, $\psi_j = \operatorname{diag}\{F_n \otimes \mathbb{I}_{l_j^1}, \dots, F_n \otimes \mathbb{I}_{l_j^{N_j}}\}$ for $j = \{z, v\}$ and the sets $\mathcal{S}_j$ restrict each of the segments of $\tilde{z}$ and $\tilde{v}$ to the pattern of complex conjugates from Corollary 1, is equivalent to (10) in the sense that

$$z^* = \psi_z \tilde{z}^*, \quad v^* = \psi_v \tilde{v}^*, \tag{12a}$$

$$\gamma^* = \psi_v \tilde{\gamma}^*, \quad \bar{\lambda}^* = \tilde{\bar{\lambda}}^*, \quad \underline{\lambda}^* = \tilde{\underline{\lambda}}^*, \tag{12b}$$

where $(z, v, \gamma, \bar{\lambda}, \underline{\lambda})^*$ and $(\tilde{z}, \tilde{v}, \tilde{\gamma}, \tilde{\bar{\lambda}}, \tilde{\underline{\lambda}})^*$ are primal and dual optimizers for (10) and (11), respectively.

Now that CBCQP (10) has been block diagonalized, the following theorem uses the pattern of complex conjugates from Corollary 1 in order to truncate problem (11).

Theorem 3 (Truncation of CBCQP [18]): Given a CBCQP of order n, which has been decomposed according to Theorem 2, then the following CBCQP,

$$\min \; \frac{1}{2} Re(\hat{z}^H \hat{J} \hat{z}) + Re(\hat{q}^H \hat{z}) \tag{13a}$$

$$s.t. \;\; \operatorname*{aug}_{N_v} \{\hat{K} \hat{z} - \hat{v}\} = 0 \tag{13b}$$

$$\underline{v} \leq \psi_v \operatorname*{aug}_{N_v} \{\hat{v}\} \leq \bar{v}, \tag{13c}$$

where $\hat{q} = \operatorname{trunc}_{N_z} \tilde{q}$, $\hat{J} = \operatorname{trunc}_{N_z} \tilde{J}$ and $\hat{K} = \operatorname{trunc}_{N_v} \tilde{K}$, is equivalent to (11) in the sense that

$$\hat{z}^* = \operatorname*{trunc}_{N_z} \tilde{z}^*, \quad \hat{v}^* = \operatorname*{trunc}_{N_v} \tilde{v}^*, \tag{14a}$$

$$\hat{\gamma}^* = \tilde{\gamma}^*, \quad \hat{\bar{\lambda}}^* = \tilde{\bar{\lambda}}^*, \quad \hat{\underline{\lambda}}^* = \tilde{\underline{\lambda}}^*, \tag{14b}$$

where $(\hat{z}, \hat{v}, \hat{\gamma}, \hat{\bar{\lambda}}, \hat{\underline{\lambda}})^*$ and $(\tilde{z}, \tilde{v}, \tilde{\gamma}, \tilde{\bar{\lambda}}, \tilde{\underline{\lambda}})^*$ are primal and dual optimizers for (13) and (11), respectively.

B. ADMM for Block Circulant MPC

The version of Algorithm 1 for a CBCQP, or equivalently a BCMPC problem, is outlined in Algorithm 2 and the individual steps are presented in the following paragraphs. Before entering its main loop, Algorithm 2 requires the mapping of q into the complex Fourier domain. Next, the algorithm solves (SP1) projected onto the Fourier domain,

Algorithm 2 ADMM for Block Circulant MPC

```
Input: State $x(t)$
Output: Input $u(t)$
Set $x_0^0 = x(t)$ and $\hat{y}^0, \hat{\gamma}^0 = 0$; compute $\underline{v}, \bar{v}$ and $\hat{q}$
for $i = 1$ to $i_{max}$ do
    Update $\hat{z}^i$ using (S$\hat{\text{P}}$1)
    Update $\hat{v}^i$ using (S$\hat{\text{P}}$2)
    Update $\operatorname{trunc}_{N_v} \hat{\gamma}^i$ using (S$\hat{\text{P}}$3)
    if $\| \operatorname{aug}_{N_v} \{\hat{v}^i - \hat{v}^{i-1}\} \|_2^2 < \epsilon$ and
            $\|\hat{\gamma}^i - \hat{\gamma}^{i-1}\|_2^2 < \epsilon$ then
        break
    end if
end for
return $u(t) = \psi_u \operatorname{aug}\{(\hat{z}_0, \dots, \hat{z}_{\frac{n-1}{2} n_u})^{\mathrm{T}}\}$
```

$$\left(\hat{J} + \rho \hat{K}^H \hat{K} \right) \hat{z}^i = \hat{K}^H (\rho \hat{v}^{i-1} - \operatorname*{trunc}_{N_v} \hat{\gamma}^{i-1}) - \hat{q}. \tag{S$\hat{\text{P}}$1}$$

When subproblem (SP1) is projected onto the Fourier domain as in (S$\hat{\text{P}}$1), it is simplified in two ways. On one hand, the block diagonalized matrices have been reduced to a maximum of $Tn(\max\{n_u, pn_y\})^2$ nonzero elements, where T is the prediction horizon of the MPC problem. On the other, the pattern of complex conjugates allows vectors and matrices to be truncated as in Theorem 2.

The modified subproblem (SP2) reads as

$$\hat{v}^i = \operatorname*{trunc}_{N_v} \{ \psi_v^H \operatorname*{sat}_{[\underline{v}, \bar{v}]} \left\{ \psi_v \operatorname*{aug}_{N_v} \{ \hat{K} \hat{z}^i + \rho^{-1} \operatorname*{trunc}_{N_v} \hat{\gamma}^{i-1} \} \right\} \}. \tag{S$\hat{\text{P}}$2}$$

The discrete Fourier transformations required in subproblem (S$\hat{\text{P}}$2) are the main drawbacks of algorithm (2) and the problem sizes required to outperform algorithm (1) are discussed in Section IV-C.

The decomposed and truncated dual variables are updated using

$$\operatorname*{trunc}_{N_v} \hat{\gamma}^i = \operatorname*{trunc}_{N_v} \hat{\gamma}^{i-1} + \rho(\hat{K} \hat{z}^i - \hat{y}^i). \tag{S$\hat{\text{P}}$3}$$

As mentioned in the proof of Theorem 3, $\hat{\gamma} \in \mathcal{S}_v$ and it is therefore sufficient to update the truncated dual variable. In practice, the term $\hat{K}\hat{z}^i$ can be cached during $(\mathrm{S\hat{P}2})$ and reused in $(\mathrm{S\hat{P}3})$.

Finally, convergence of Algorithm 2 is checked via the condition

$$\| \underset{N_v}{\operatorname{aug}}\{\hat{v}^i - \hat{v}^{i-1}\}\|_2^2 < \epsilon, \tag{15}$$

$$\|\hat{\gamma}^i - \hat{\gamma}^{i-1}\|_2^2 < \epsilon. \tag{16}$$

To see that this criterion is equivalent to the one of Algorithm 1, note that $\| \operatorname{aug}_{N_v}\{\hat{v}\}\|_2 = \|\psi_v^H v\|_2 = \|v\|_2$.

As shown in Theorems 2 and 3, Algorithms 1 and 2 produce an equivalent solution in the sense that the minimizers z^* and $\hat{z}^*$ are related by $z^* = \psi_z \operatorname{aug}_{N_z}\{\hat{z}^*\}$.

C. Computational Complexity

It is assumed that n is even, that the matrices in Definition 5 are full and that $J \in \mathbb{R}^{(Nn_zn)\times(Nn_zn)}$ and $K \in \mathbb{R}^{(Nn_vn)\times(Nn_zn)}$ are partitioned into segments of identical lengths n_z and n_v, respectively, with $n_v = n_z$. Algorithm 2 makes use of FFTs during initialization, finalization and in subproblem $(\mathrm{S\hat{P}2})$. We assume that the complexity of an r-point FFT is $\mathcal{O}\left(r \log r\right)$ and also define $L := Nn_z$.

For both algorithms, the main burden lies in solving the linear system in subproblems (SP1) and $(\mathrm{S\hat{P}1})$. Depending on the structure of the linear system, it can be solved in numerous ways. For simplicity, it is assumed that the linear system is solved using a matrix inverse that is pre-computed offline. In that case, (SP1) is of complexity $\mathcal{O}\left((nL)^2\right)$, while the block diagonalization and truncation in $(\mathrm{S\hat{P}1})$ reduce the complexity to $\mathcal{O}\left(4(n_c{+}1)L^2\right)$, where the factor 4 accounts for the required complex arithmetic and n_c is defined in Corollary 1. Because the operations of the saturation function are negligible, (SP2) accounts for a complexity of $\mathcal{O}\left((nL)^2{+}nL\right)$. Note that the term Kz^i is reused in (SP3). As the projection onto the boundaries $[\underline{v}, \bar{v}]$ must be carried out in the original domain, the drawbacks of the Fourier transformation become evident in $(\mathrm{S\hat{P}2})$. This results in a complexity of $\mathcal{O}\left((n_c{+}1)(4L^2{+}2L){+}2Ln\log n\right)$. Lastly, (SP3) and $(\mathrm{S\hat{P}3})$ are of complexities $\mathcal{O}\left(2nL\right)$ and $\mathcal{O}\left(4(n_c{+}1)L\right)$, respectively.

Algorithms 1 and 2 have an overall complexity of $\mathcal{O}\left(nL(2nL{+}3)\right)$ and $\mathcal{O}\left((n_c{+}1)(8L^2{+}6L){+}2Ln\log n\right)$, respectively. Under the assumptions of this section, Algorithm 2 is more efficient than 1 if $L(n^2{-}2n{-}4) > 3{+}n\log n$.

V. Simulations

A. Random Constrained QP with Block Circulant Blocks

We first consider a set of randomly generated CBCQPs to gauge performance of Algorithm 2. Figure 1B shows the decomposed left-hand side of (SP1), $J + \rho K^{\mathrm{T}}K$, of a QP of order $n = 4$ before applying the truncation. On one hand, it is evident how the transformation matrices ψ_z and ψ_v block diagonalize $J + \rho K^{\mathrm{T}}K$. On the other, the pattern of complex conjugate blocks motivating the truncation becomes apparent. Figure 1A shows the execution times of Algorithms 1 and 2 for random block circulant QPs of increasing order n with $N = 1$ and $l_z = l_v = 10$. Even though the analysis of Section IV-C accounted for the required Fourier transformations, Figure 1A reveals that for small n the additional operations required in $(\mathrm{S\hat{P}2})$, such as truncating, permuting and augmenting the vectors, are not negligible. For larger n these side effects lose their significance and the superiority of Algorithm 2 becomes evident. Additional simulations can be found in [18].

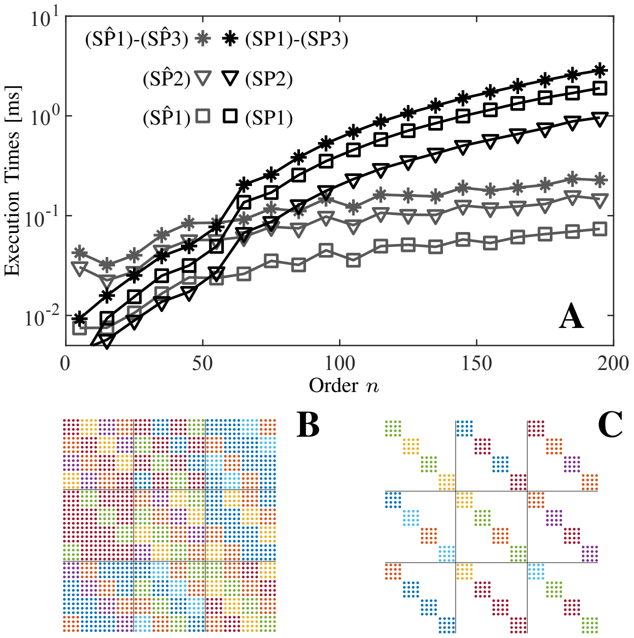

Fig. 1: **(A)** Logarithmically scaled average execution times of the different steps of Algorithms 1 (black) and 2 (gray) in milliseconds for random block circulant QPs. **(B, C)** Sparsity patterns of **(B)** $J + \rho K^{\mathrm{T}}K$ and **(C)** $\tilde{J} + \rho\tilde{K}^H\tilde{K}$ for $N_z = N_v = 3$ and $l_z^j = l_v^j = 4$, $j = 1, 2, 3$. The colors are proportional to the magnitude of the matrix elements.

B. Traffic Flow Control using block circulant MPC

We consider the regulation of the traffic flow on either a circular road with $n-1$ human-driven vehicles and one autonomous vehicle (AV) or an infinite road with n-periodic dynamics or a periodically extended road of finite length with $(n-2)/2$ vehicles [6]. The dynamics of each vehicle are assumed to be identical and modeled by a discrete-time second-order system. The states of each vehicle represent the velocity and the distance to the preceding vehicle. After linearization around a desired set-point, the system with $2n$ states and one controllable input – the acceleration of the AV – can be reformulated as a partially BC linear system [6]. An MPC problem is then defined as in (1) with a horizon $T = 40$ and a sampling time of $0.05s$. The acceleration of the AV is constrained and it is assumed that the AV has access to an exact prediction of a disturbance, e.g. in case

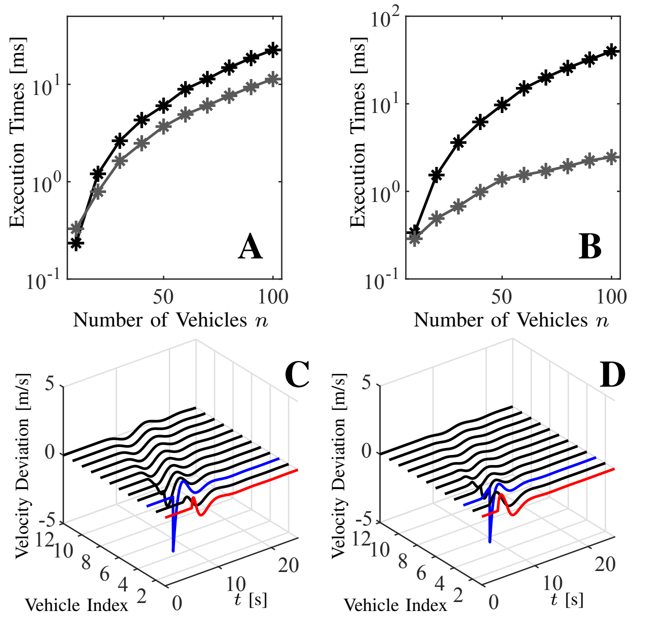

Fig. 2: **(A, B)** Logarithmically scaled average execution times for one iteration of Algorithms 1 (black) and 2 (gray) in milliseconds for the traffic flow control example with **(A)** one and **(B)** n controllable vehicles. **(C, D)** Response profiles of $n = 20$ vehicles to a perturbation impacting vehicle 2 (blue). Only 12 vehicles are shown. **(C)** The AV (vehicle 1, red) is controlled using LQR as in [6]. **(D)** The AV is controlled using Algorithm 2 with a disturbance prediction.

of a foreseeable lane change. Note that for this example the input matrix B in (1b) is not BC and Algorithm 2 has to be modified accordingly, i.e. $\psi_{z,v} = (\mathbb{I}_T \otimes F_n \otimes I_2) \oplus I_T$ and $z = (x_1, \dots, x_T, u_0, \dots, u_{T-1})^{\mathrm{T}}$ in Theorem 2.

Figure 2A compares the performance of Algorithm 1 and 2 solving the partially BCMPC problem with random feasible initial conditions for an increasing number of AVs n. Figure 2B compares their performance when all vehicles are controllable. Each problem was solved with an average number of 100 ADMM iterations. As for the example of Section V-A, Algorithm 2 performs worse than Algorithm 1 for small n when the drawbacks ($\mathrm{S\hat{P}2}$) and other side effects outweigh the computational gains in ($\mathrm{S\hat{P}1}$). Even though the CQP in Figure 2A can not be fully decomposed, the benefits of the Fourier transformation become evident for larger n.

Figures 2C and 2D depict the velocity profiles of the initially unstable system stabilized by a linear quadratic regulator (LQR) [6] and Algorithm 2, respectively. In addition to its ability to consider state and input constraints, Algorithm 2 outperforms the LQR by exploiting the disturbance prediction.

VI. Conclusions

This paper demonstrated how to exploit the particular structure of an MPC problem for BC systems. Based on the properties of BC matrices, a BCMPC problem was defined and connected to a general constrained QP with BC blocks. A transformation was derived which block diagonalizes any constrained QP with BC blocks and allows to truncate the transformed vectors. A modified ADMM algorithm for the transformed and truncated system was developed. The modified ADMM algorithm was tested using a series of random constrained QPs with BC problem data and a traffic flow control problem. In both cases, the evaluation of the results revealed that the modified ADMM algorithm performs significantly better for increasing problem sizes.

References

[1] J. L. Jerez, P. J. Goulart, S. Richter, G. A. Constantinides, E. C. Kerrigan, and M. Morari, "Embedded online optimization for model predictive control at megahertz rates," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3238–3251, August 2014.

[2] C. Danielson, "An alternating direction method of multipliers algorithm for symmetric MPC," in *2018 6th IFAC Nonlinear Model Predictive Control Conference (NMPC)*, Madison, Wisconsin, August 2018, pp. 319–324.

[3] P. J. Davis, *Circulant Matrices*. Wiley-Interscience, 1979.

[4] R. D'Andrea and G. E. Dullerud, "Distributed control design for spatially interconnected systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 9, pp. 1478–1495, September 2003.

[5] N. Denis and D. Looze, "H∞ controller design for systems with circulant symmetry," in *1999 38th IEEE Conference on Decision and Control (CDC)*, Phoenix, Arizona, December 1999, pp. 3144–3149.

[6] Y. Zheng, J. Wang, and K. Li, "Smoothing traffic flow via control of autonomous vehicles," *arXiv e-prints*, p. arXiv:1812.09544, December 2018.

[7] D. L. Laughlin, M. Morari, and R. D. Braatz, "Robust performance of cross-directional basis-weight control in paper machines," *Automatica*, vol. 29, no. 6, pp. 1395–1410, November 1993.

[8] S. Gayadeen and S. R. Duncan, "Uncertainty modeling and robust stability analysis of a synchrotron electron beam stabilisation control system," in *2012 51st IEEE Conference on Decision and Control (CDC)*, Maui, Hawaii, December 2012, pp. 931–936.

[9] R. W. Brockett and J. L. Willems, "Discretized partial differential equations: Examples of control systems defined on modules," *Automatica*, vol. 10, no. 5, pp. 507 – 515, 1974.

[10] P. Massioni and M. Verhaegen, "Distributed control for identical dynamically coupled systems: A decomposition approach," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 124–135, January 2009.

[11] J.-H. Li, S.-Z. Zhao, J. Zhao, and Y. ping Li, "Stability analysis for circulant systems and switched circulant systems," in *2004 43rd IEEE Conference on Decision and Control (CDC)*, Nassau, Bahamas, December 2004, pp. 2805–2809.

[12] P. Massioni and M. Verhaegen, "Subspace identification of circulant systems," *Automatica*, vol. 44, no. 11, pp. 2825–2833, November 2008.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, January 2011.

[14] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.

[15] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *arXiv e-prints*, p. arXiv:1711.08013, November 2017.

[16] D. J. Rose, "Matrix identities of the fast Fourier transform," *Linear Algebra and its Applications*, vol. 29, pp. 423–443, February 1980.

[17] S. Bittanti, A. Laub, and J. Willems, *The Riccati Equation*, ser. Communications and Control Engineering. Springer Berlin Heidelberg, 1991.

[18] I. Kempf, P. J. Goulart, and S. Duncan, "ADMM for block circulant model predictive control," *arXiv e-prints*, p. arXiv:1904.04115, April 2019.