2022 European Control Conference (ECC)
July 12-15, 2022. London, United Kingdom

Model Predictive Control for Electron Beam Stabilization in a Synchrotron

Idris Kempf*, Paul J. Goulart*, Stephen R. Duncan* and Michael Abbott**

Abstract— **Electron beam stabilization in a synchrotron is a large-scale disturbance rejection problem, with hundreds of inputs and outputs, that is sampled at frequencies higher than 10 kHz. In this feasibility study, we focus on the practical issues of implementing model predictive control (MPC) for the heavily ill-conditioned plant of the electron beam stabilization problem at Diamond Light Source. If the terminal cost matrix of the MPC problem is obtained from the discrete-time algebraic Riccati equation (DARE), the ill-conditioned plant results in an ill-conditioned Hessian. Here, we propose obtaining a stabilizing terminal cost matrix from a modified version of the DARE that includes a constraint on the condition number of the Hessian. We implement our control algorithm on the hardware designated for the imminent Diamond Light Source upgrade, and show that the modified terminal cost matrix allows MPC to be executed at the rate required for synchrotron control. MPC overcomes various problems of standard electron beam stabilization techniques, and we show that the successful implementation can increase the stability of photon beams in synchrotron light sources.**

Index Terms— **Model predictive control, fast gradient method, embedded systems, synchrotron**

I. Introduction

A synchrotron light source is a type of particle accelerator in which charged particles, typically electrons, travel around a circular path called the *storage ring*. When the electrons' paths are bent around the storage ring at relativistic speeds, they lose kinetic energy and emit it in the form of exceptionally bright synchrotron light, which is used for various experiments. An assembly of magnets produces a magnetic field that confines the electrons in the storage ring. Large magnets steer and focus the electron beam whilst smaller *corrector magnets* attenuate vibrations induced by disturbances and reduce the trajectory error of the electrons down to a few $\mu\mathrm{m}$. These disturbances are caused by internal devices, such as the beam light extraction devices, or transmitted through the girders on which the magnet arrays are attached. The position of the electron beam is measured using *beam position monitors* (BPMs) and the corrector magnets are controlled in a feedback loop that is sampled at a rate typically between $10\,\mathrm{kHz}$ and $100\,\mathrm{kHz}$. The beam trajectory error must be minimized in order to produce high brilliance synchrotron light. This control system is referred to as *fast orbit feedback* and typically has a few hundred BPMs (outputs) and few hundred corrector magnets (inputs). Diamond Light Source (DLS) is the UK's national synchrotron facility, and its $560\,\mathrm{m}$ circumference storage ring accommodates over 20 experimental stations. DLS has completed the conceptual design phase of a significant upgrade (DLS-II), which will increase the brightness of the synchrotron light by raising the electron beam energy from $3\,\mathrm{GeV}$ to $3.5\,\mathrm{GeV}$ [1] and the number of sensors from 172 to 252 and the number of actuators from 173 to 396, which significantly increases the dimension of the control problem. In the current facility only one type of corrector magnet is used, but DLS-II will instead use separate types: slow correctors with strong magnetic fields for low-bandwidth correction and fast correctors with weak magnetic fields for high-bandwidth correction. In addition, the sampling frequency will be increased from $10\,\mathrm{kHz}$ to $100\,\mathrm{kHz}$.

The consequences of introducing two types of corrector magnets are twofold. First, the widely used *modal decomposition* [2] that diagonalizes the input-output transfer function matrix using a singular value decomposition can no longer be applied. Most synchrotrons [3]–[6] implement two separate feedback controllers instead, which requires introduction of a frequency-deadband or an alternative compensation to avoid cross-talk between the controllers. Second, amplitude and slew-rate actuator constraints must be considered, which is exacerbated by the ill-conditioned plant. Odd placements of sensors and magnets produce output directions that are difficult to control, which can cause large corrector gains or even lead to instabilities [7].

Model predictive control (MPC) allows for an arbitrary number of actuator arrays and provides a systematic way to handle actuator constraints while achieving the same or better disturbance attenuation [8] as linear control methods. However, the MPC algorithm uses real-time optimization and considerably increases the computational complexity of the fast orbit feedback system. Formulating MPC for the electron beam stabilization problem results in a constrained quadratic program with hundreds of decision variables, and the highly ill-conditioned plant can negatively affect the convergence properties of the algorithm. A tailored MPC implementation is therefore required to obtain an MPC scheme that operates at frequencies higher than $10\,\mathrm{kHz}$. In anticipation of the upcoming DLS-II upgrade, it was decided to assess the feasibility and performance of installing MPC on the existing DLS-I storage ring. This paper describes an assessment of the design and conception of the future DLS-II fast orbit feedback architecture and allows for optimal dimensioning of the required controller hardware.

The paper is organized as follows. We use standard

*Corresponding author: idris.kempf@eng.ox.ac.uk. The 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.

**Diamond Light Source, Didcot, UK.

978-3-907144-07-7 ©2022 EUCA

modeling techniques for setpoint tracking and observer design, but include these details in Section II for the benefit of practitioners in the synchrotron community who may be unfamiliar with these methods. We formulate our MPC problem in Section III, which we solve using the fast gradient method. The conditioning of the Hessian is investigated in Section IV. Finally, Section V details the parallel implementation of MPC on a multicore digital signal processor (DSP).

II. Preliminaries

A. State-Space Model

For DLS-II, the relationship between the $n_y = 252$ beam displacements $\mathbf{y}_k \in \mathbb{R}^{n_y}$ measured around the ring and the $n_u = 396$ corrector magnets inputs $\mathbf{u}_k \in \mathbb{R}^{n_u}$ at time $t = kT_s$ can be modeled as

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \qquad \mathbf{y}_k = \mathbf{C}\mathbf{x}_{k-\mu} + \mathbf{d}_k, \tag{1}$$

where $T_s = 10\,\mu\mathrm{s}$ is the sampling time, $\mu = 10$ is the measurement delay in terms of time steps, $\mathbf{x}_k \in \mathbb{R}^{n_x}$ with $n_x = n_u$ are the states and $\mathbf{d}_k \in \mathbb{R}^{n_y}$ the disturbances. The diagonal matrices $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$ and $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$ model the actuator dynamics. The matrix $\mathbf{C} \in \mathbb{R}^{n_y \times n_x}$ is called the *orbit response matrix* and typically has a condition number on the order of 10^4 [7].

The inputs $\mathbf{u}_k$ are subjected to amplitude and slew-rate constraints that can be modeled as

$$\mathcal{U}(\mathbf{u}_{k\text{-}1}) := \left\{ \mathbf{u}_k \in \mathbb{R}^{n_u} \;\middle|\; \begin{matrix} \text{-}\alpha \leq \mathbf{u}_k \leq \alpha, \\ \text{-}\rho \leq \mathbf{u}_k \text{ - } \mathbf{u}_{k\text{-}1} \leq \rho \end{matrix} \right\}, \tag{2}$$

where the inequalities are to be read component-wise and $\mathbf{u}_{k\text{-}1}$ is the input applied at time $t = (k-1)T_s$. The limits $\alpha \in \mathbb{R}^{n_u}$ and $\rho \in \mathbb{R}^{n_u}$ depend on the types of corrector magnets. We will assume symmetric limits on both slew-rate and amplitude, but our results are easily modified to allow asymmetric limits.

In addition to the plant model (1), the observer, which is introduced in Section II-B, requires a disturbance model. The disturbances are modeled using a first-order system that is driven by zero-mean independent and identically distributed white noise [9], i.e.

$$\mathbf{d}_{k+1} = \mathbf{A}_d \mathbf{d}_k + \mathbf{v}_k, \tag{3}$$

where $\mathbf{v}_k \sim \mathcal{N}(0, \sigma_\mathbf{v}^2)$ and we choose $\mathbf{A}_d = I$. Alternatively, the matrix $\mathbf{A}_d$ can be obtained from a first-order autoregressive fit from measurement data.

Note that the vector $\mathbf{y}_k$ describes the displacement in either the horizontal or vertical direction perpendicular to the motion of the electron beam. These directions are independent, and the electron beam stabilization problem includes two separate systems of the form of (1).

B. Observer

Standard methods from electron beam stabilization use output feedback to control the input-output representation of (1) [7], whereas MPC uses state feedback. The vectors $\mathbf{x}_k$ and $\mathbf{d}_k$ are not measurable and must be inferred from $\mathbf{y}_k$ using an observer.

The measurements are delayed by μ time steps and the incoming measurement $\mathbf{y}_k$ at time $t = kT_s$ contains information about the state $\mathbf{x}_{k\text{-}\mu}$ at time $t = (k-\mu)T_s$. One way to integrate the delayed measurements is to formulate a delay-free system by augmenting (1) with $\mu \times n_x$ states, i.e. defining the augmented state vector $\mathbf{z}_k := (\mathbf{x}_k^\mathrm{T}, \mathbf{x}_{k-1}^\mathrm{T}, \dots, \mathbf{x}_{k-\mu}^\mathrm{T})^\mathrm{T}$. An observer for (1) and (3) can then be formulated as

$$\begin{pmatrix} \hat{\mathbf{z}}_{k+1} \\ \hat{\mathbf{d}}_{k+1} \end{pmatrix} = \begin{bmatrix} \mathbf{A}_\text{aug} & 0 \\ 0 & \mathbf{A}_d \end{bmatrix} \begin{pmatrix} \hat{\mathbf{z}}_k \\ \hat{\mathbf{d}}_k \end{pmatrix} + \begin{bmatrix} \mathbf{B}_\text{aug} \\ 0 \end{bmatrix} \mathbf{u}_k + \mathbf{L}\Delta\hat{\mathbf{y}}_k, \tag{4}$$

where variables with a hat denote estimated quantities, $\mathbf{L} \in \mathbb{R}^{(\mu+1)n_x + n_y \times n_y}$ is the Kalman filter steady-state gain [10, Ch. 7.3], $\mathbf{A}_\text{aug}$ and $\mathbf{B}_\text{aug}$ are defined in [11, Eq. 9] and

$$\Delta\hat{\mathbf{y}}_k := \mathbf{y}_k - \mathbf{C}_\text{aug}\hat{\mathbf{z}}_k - \hat{\mathbf{d}}_k = \mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_{k-\mu} - \hat{\mathbf{d}}_k.$$

The observer requires a matrix-vector multiplication with the dense matrix $\mathbf{L}$, which is a computationally expensive operation that can be avoided as follows. First, partition the observer gain as $\mathbf{L} = [\mathbf{L}_0, \dots \mathbf{L}_\mu^\mathrm{T}, \mathbf{L}_d^\mathrm{T}]^\mathrm{T}$, where the partitioning of $\mathbf{L}$ matches the partitioning of $\mathbf{z}_k$ and $\mathbf{d}_k$. Then, update the most delayed state $\hat{\mathbf{x}}_{k\text{-}\mu}$ and the disturbance estimate $\hat{\mathbf{d}}_k$ using $\mathbf{L}_\mu$ and $\mathbf{L}_d$, respectively, and reserve $\mathbf{L}_\mu\Delta\hat{\mathbf{y}}_k$. Finally, update the states $\hat{\mathbf{x}}_{k\text{-}i}$, $i = 0, \dots, \mu\text{ - }1$, by adding $\mathbf{A}^{\mu\text{-}i}\mathbf{L}_\mu\Delta\hat{\mathbf{y}}_k$. Note that the matrices $\mathbf{A}^i$ are diagonal and can be precomputed offline.

C. Setpoint Calculation

The aim of the control system is to reject the disturbances $\mathbf{d}_k$ in (1). In response to a constant disturbance, a zero steady-state output $\mathbf{y}_k$ requires the open-loop transfer function to have integrating behavior [12]. Because (1) lacks integrating behavior, the controller must implement the integrator. One way is to augment the system with a set of output integrators. However, this method would slow down the subsequent MPC algorithm by increasing the number of optimization variables. Alternatively, one can compute the setpoints $\bar{\mathbf{u}}$ and $\bar{\mathbf{x}}$ and use the feedback law $\mathbf{u}_k = \bar{\mathbf{u}} + \mathbf{u}_k^\star$ [9], where $\mathbf{u}_k^\star$ is obtained as the solution to the MPC problem. The setpoints should be calculated such that $\lim_{k\to\infty} \mathbf{y}_k = 0$, which using (1) yields

$$\begin{pmatrix} 0 \\ \hat{\mathbf{d}}_k \end{pmatrix} = \begin{bmatrix} I - \mathbf{A} & -\mathbf{B} \\ -\mathbf{C} & 0 \end{bmatrix} \begin{pmatrix} \bar{\mathbf{x}}_k \\ \bar{\mathbf{u}}_k \end{pmatrix} =: \mathbf{S} \begin{pmatrix} \bar{\mathbf{x}}_k \\ \bar{\mathbf{u}}_k \end{pmatrix}. \tag{5}$$

The coefficient matrix $\mathbf{S} \in \mathbb{R}^{n_u + n_y \times 2n_u}$ has more columns than rows and the Moore-Penrose pseudoinverse $\mathbf{S}^\dagger = \left(\mathbf{S}^\mathrm{T}\mathbf{S}\right)^{-1}\mathbf{S}^\mathrm{T}$ can be used to solve for $\bar{\mathbf{x}}_k$ and $\bar{\mathbf{u}}_k$. Note the zeros in the left-hand side vector of (5), so in practice, only the last n_y columns of $\mathbf{S}^\dagger$ need to be considered.

III. Model Predictive Control

A. Problem Formulation

At time $t = kT_s$, an MPC scheme computes a control input by predicting the future evolution of the system and minimizing a quadratic objective function over the planning horizon N, while considering inputs that lie in the constraint set (2) only. This can be achieved via repeated solution of the following constrained quadratic program (CQP):

$$\begin{aligned} \min_{\mathbf{x}_{k|i},\mathbf{u}_{k|i}} \; & \|\mathbf{x}_{k|N}-\bar{\mathbf{x}}\|^2_{\mathbf{P}}+\sum_{i=0}^{N\text{-}1}\big(\|\mathbf{x}_{k|i}-\bar{\mathbf{x}}\|^2_{\mathbf{Q}}+\|\mathbf{u}_{k|i}-\bar{\mathbf{u}}\|^2_{\mathbf{R}}\big) \\ \text{s.t.} \; & \\ & \left.\begin{aligned} &\mathbf{x}_{k|i+1} = \mathbf{A}\mathbf{x}_{k|i} + \mathbf{B}\mathbf{u}_{k|i} \\ &\mathbf{u}_{k|i} \in \mathcal{U}(\mathbf{u}_{k|i\text{-}1}) \end{aligned}\right\} \forall i = 0,\dots,N-1, \\ & \mathbf{x}_{k|0} = \hat{\mathbf{x}}_k,\;\; \mathbf{u}_{k|\text{-}1} = \mathbf{u}^\star_{k\text{-}1|0}, \end{aligned} \tag{6}$$

where $\mathbf{u}^\star_{k\text{-}1|0}$ is the control input at time $(k$ - $1)T_s$. Even though the solution of (6) is a sequence of inputs $\mathbf{u}^\star_{k|0},\dots,\mathbf{u}^\star_{k|N\text{-}1}$, only the first input $\mathbf{u}^\star_{k|0}$ is applied to the plant and the optimization repeated at the next time step $(k+1)T_s$. The matrices $\mathbf{Q} := \mathbf{C}^{\mathrm{T}}\mathbf{C} \succeq 0$ and $\mathbf{R} \succ 0$ are the state and output weighting matrices, respectively, while $\mathbf{P} = \mathbf{P}^{\mathrm{T}} \succeq 0$ is the terminal cost matrix.

To limit the complexity of the CQP, only input constraints are considered. Problem (6) can then be *condensed* by eliminating the state-transition equations [13], producing the equivalent problem

$$\min_{\mathbf{u}} \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{J}\mathbf{u} + \mathbf{q}(\hat{\mathbf{x}}_k,\hat{\mathbf{d}}_k)^{\mathrm{T}}\mathbf{u} \quad \text{s.t.} \quad \mathbf{u}\in\mathcal{S}(\mathbf{u}_{\text{-}1}), \tag{7}$$

where $\mathbf{u} := (\mathbf{u}^{\mathrm{T}}_{k|0},\dots,\mathbf{u}^{\mathrm{T}}_{k|N\text{-}1})^{\mathrm{T}} \in \mathbb{R}^{Nn_u}$, $\mathcal{S}(\mathbf{u}_{\text{-}1}) := \mathcal{U}(\mathbf{u}_{k|\text{-}1}) \times \cdots \times \mathcal{U}(\mathbf{u}_{k|N\text{-}2})$ and the definitions of $\mathbf{J} = \mathbf{J}^{\mathrm{T}} \in \mathbb{R}^{Nn_u\times Nn_u}$, which is referred to as the *Hessian*, and $\mathbf{q}(\hat{\mathbf{x}}_k,\hat{\mathbf{d}}_k) : \mathbb{R}^{n_x}\times\mathbb{R}^{n_y} \mapsto \mathbb{R}^{Nn_u}$ can be found in [14]. Note that the vector $\mathbf{q}(\hat{\mathbf{x}}_k,\hat{\mathbf{d}}_k)$, which is an affine function of $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{d}}_k$, and the set $\mathcal{S}(\mathbf{u}_{\text{-}1})$, are updated at every time step.

When the constraints (2) are enforced, it has been shown that MPC outperforms standard methods, such as internal model control (IMC) [8], [15]. Simulations also showed that there is little performance improvement for horizons $N > 2$, which is why we consider only $N \leq 2$ in the following.

B. Fast Gradient Method

The *fast gradient method* (FGM) and the *alternating direction method of multipliers* (ADMM) are suitable algorithms for solving (7) in real time [16]. In [17], we showed that because the projection onto the constraint set (2) can be explicitly computed, FGM outperforms ADMM in terms of computational speed. Alternatively, the solutions to problem (7) could also be precomputed offline by solving a multiparametric QP [13]. However, the dimension of our problem is far in excess of the dimension that can typically be solved via multiparametric QPs.

The FGM is summarized in Alg. 1 (lines 4–8) with a constant step size $\beta = (\lambda^{1/2}_{max} - \lambda^{1/2}_{min})/(\lambda^{1/2}_{max} + \lambda^{1/2}_{min})$, where λ_{min} and λ_{max} are the minimum and maximum eigenvalues of the Hessian $\mathbf{J}$ [18, Ch. 2.2]. On line 6, the FGM requires projection operator $\mathcal{P}_{\mathcal{S}}$ onto $\mathcal{S}(\mathbf{u}_{\text{-}1})$. For $N = 1$, the projection limits each component of $\mathbf{u}$ to a minimum and maximum value given by (2). For $N = 2$, the projection consists of projecting the pairs $(\mathbf{u}^i_0,\mathbf{u}^i_1)$, where i denotes the ith actuator, onto a hexagon [17]. Note that on line 3, we warm-start by initializing the FGM using the input calculated at time $(k-1)T_s$, which improves the convergence properties of the algorithm [19]. Lines marked with the circled arrows ↻ denote synchronization steps of the parallel implementation described in Section V.

For the practical implementation, we choose a fixed number of iterations I_{max} rather than using a stopping criterion. For a desired solution accuracy, I_{max} can be calculated and increases as $\kappa(\mathbf{J})$ does [19]. A small $\kappa(\mathbf{J})$ is therefore essential to execute Alg. 1 within the very tight time constraints of synchrotrons.

Algorithm 1 MPC for electron beam stabilization

```
Input: $\mathbf{y}_k$
Output: $\mathbf{u}_k \hookleftarrow \mathbf{p}_{I_{max}}$
1: Update observer $\Rightarrow \hat{\mathbf{x}}_k, \hat{\mathbf{d}}_k$                ↻ ↻
2: Update $\mathbf{q} = \mathbf{q}(\hat{\mathbf{x}}_k,\hat{\mathbf{d}}_k)$ and $\mathcal{S} = \mathcal{S}(\mathbf{u}_{k\text{-}1})$      ↻
3: Set $\mathbf{v}_i = \mathbf{u}_{k\text{-}1}$ and $\mathbf{p}_i = 0$
4: for $i = 0$ to $I_{max} - 1$ do
5:     $\mathbf{t}_i = (I - \mathbf{J}\lambda^{\text{-}1}_{max})\mathbf{v}_i - \mathbf{q}\lambda^{\text{-}1}_{max}$              ↻
6:     $\mathbf{p}_{i+1} = \mathcal{P}_{\mathcal{S}}(\mathbf{t}_i)$
7:     $\mathbf{v}_{i+1} = (1+\beta)\mathbf{p}_{i+1} - \beta\mathbf{p}_i$
8: end for                                                          ↻
```

IV. Conditioning of the Hessian

The optimization problem (6) has a unique solution if $\mathbf{R} \succ 0$, $\mathbf{Q} \succeq 0$ and if the pairs $(\mathbf{A},\mathbf{B})$ and $(\mathbf{A},\mathbf{Q}^{\frac{1}{2}})$ are controllable and observable, respectively [13, Ch. 12]. Because (1) is stable and there are no state constraints, the MPC scheme is guaranteed to be stable if [13]

$$\mathbf{P} \preceq \mathbf{A}^{\mathrm{T}}\mathbf{P}\mathbf{A} - \mathbf{A}^{\mathrm{T}}\mathbf{P}\mathbf{B}\left(\mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{B}+\mathbf{R}\right)^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{A} + \mathbf{Q}, \tag{8}$$

where we choose the matrices $\mathbf{Q}$ and $\mathbf{R}$ to be the same as in (6). If equality is substituted for the inequality in (8), the matrix $\mathbf{P}$ is the solution of the *discrete-time algebraic Riccati equation* (DARE). If the middle term on the right-hand side of (8) is not present, it has been shown that the solution of the DARE is ill-conditioned if $\kappa(\mathbf{Q}) \gg 1$ [20]. For the electron beam stabilization problem, $\kappa(\mathbf{Q}) \approx 10^7$ and we have observed that for reasonable choices of $\mathbf{R}$, the resulting $\mathbf{P}$ is similarly ill-conditioned.

For $N \leq 2$, the Hessian is obtained as

$$N\!=\!1: \quad \mathbf{J} = \mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{B} + \mathbf{R}, \tag{9a}$$

$$N\!=\!2: \quad \mathbf{J} = \begin{bmatrix} \mathbf{B} & 0 \\ \mathbf{AB} & \mathbf{B} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{Q} & 0 \\ 0 & \mathbf{P} \end{bmatrix} \begin{bmatrix} \mathbf{B} & 0 \\ \mathbf{AB} & \mathbf{B} \end{bmatrix} + \begin{bmatrix} \mathbf{R} & 0 \\ 0 & \mathbf{R} \end{bmatrix}. \tag{9b}$$

The following proposition relates $\kappa(\mathbf{P})$ and $\kappa(\mathbf{J})$.

Proposition 1: Suppose that $\kappa(\mathbf{P}) \gg 1$, $\lambda_{\max}(\mathbf{P}) \gg 1$ and $\|\mathbf{P}\|_2 \gg \|\mathbf{B}^{\text{-T}}\mathbf{R}\mathbf{B}\|_2$. Then $\kappa(\mathbf{J}) \gg 1$.

Proof: Using the Courant-Fischer theorem [21, Thm. 4.2.6], it holds for $N = 1$ that

$$\kappa(\mathbf{J}) = \frac{\max_{\|\mathbf{x}\|_2=1} \mathbf{x}^{\mathrm{T}}\mathbf{J}\mathbf{x}}{\min_{\|\mathbf{x}\|_2=1} \mathbf{x}^{\mathrm{T}}\mathbf{J}\mathbf{x}} \approx \frac{\lambda_{\max}(\mathbf{P})}{\min_{\|\mathbf{x}\|_2=1} \mathbf{x}^{\mathrm{T}}(\mathbf{P}+\mathbf{B}^{\text{-T}}\mathbf{R}\mathbf{B})\mathbf{x}}.$$

It follows that for reasonable choices of $\mathbf{R}$, $\kappa(\mathbf{J}) \gg 1$. The proof for $N = 2$ is similar. ■

If $\mathbf{P}$ is obtained from the DARE with $\mathbf{R} = I$, then $\kappa(\mathbf{J}) \geq 6000$, which is too large for Alg. 1 to converge to acceptable accuracy at $10\,\text{kHz}$. Increasing the magnitude of $\mathbf{R}$ reduces $\kappa(\mathbf{J})$, but in turn also worsens the performance of Alg. 1. Standard literature suggests to minimize the condition number by diagonal scaling [22, Ch. 3.1], but for the present problem, diagonal scaling leads to a reduction of $1\,\%$ to $2\,\%$ only. Instead, we propose to obtain $\mathbf{P}$ with $\kappa(\mathbf{P}) \leq \gamma$ from

$$\begin{aligned} \max_{\mathbf{P},\mu} \quad & \operatorname{trace}(\mathbf{P}) \\ \text{s.t.} \quad & \begin{bmatrix} \mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{B}+\mathbf{R} & \mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{A} \\ \mathbf{A}^{\mathrm{T}}\mathbf{P}\mathbf{B} & \mathbf{A}^{\mathrm{T}}\mathbf{P}\mathbf{A}+\mathbf{Q}-\mathbf{P} \end{bmatrix} \succeq 0, \qquad \text{(10a)} \\ & \mu I \preceq \mathbf{J} \preceq \gamma\mu I, \qquad \text{(10b)} \end{aligned}$$

where $\mathbf{P} = \mathbf{P}^{\mathrm{T}} \succeq 0$, $\mu > 0$ and (9a) or (9b) is substituted for $\mathbf{J}$. Constraint (10a) is equivalent to (8) after reformulation using the Schur complement; any $\mathbf{P}$ obtained from (10) guarantees the stability of (6). Constraint (10b) ensures that $\kappa(\mathbf{J}) \leq \gamma$ [22, Ch. 3.3]. Note that if (10b) is omitted, the optimal solution of (10) is the unique stabilizing solution of the DARE [23]. For given $\mathbf{Q}$ and $\mathbf{R}$, problem (10) returns the "best" stabilizing $\mathbf{P}$ that satisfies the condition number constraint (10b).

For $N > 2$, the Hessian $\mathbf{J}$ contains $\mathbf{Q}$ with $\kappa(\mathbf{Q}) \approx 10^8$, which complicates (10b) because $\mathbf{Q}$ is not an optimization variable. In problem (10), we therefore substitute $\mathbf{V}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$ for $\mathbf{Q}$, where $\mathbf{\Sigma} \succeq 0$ is a (diagonal) optimization variable, and modify the objective function as

$$\max_{\mathbf{P},\mu,\mathbf{\Sigma}} \qquad \operatorname{trace}(\mathbf{P}) - \alpha\|\mathbf{\Sigma}_0 - \mathbf{\Sigma}\|_{\mathrm{F}}, \qquad \text{(11)}$$

for some $\alpha > 0$ and where $\mathbf{\Sigma}_0$ contains the singular values of $\mathbf{C}$. With these substitutions, a matrix $\mathbf{Q}$ is found that is nearest to $\mathbf{C}^{\mathrm{T}}\mathbf{C}$ and with which a stabilizing $\mathbf{P}$ is obtained that satisfies (10b). Intuitively, we expect that the optimization decreases large singular values of $\mathbf{C}$ and increases small values, which we can enforce by adding the constraint $\min(\mathbf{\Sigma}_0) \preceq \mathbf{\Sigma} \preceq \max(\mathbf{\Sigma}_0)$. Otherwise, the outcome is sensitive to the choice of α in (11).

For $N = 1$, problem (10) is feasible for $\mathbf{R} = I$ and $\gamma \in \{10, 20, 50\}$ and the required number of iterations for Alg. 1 is illustrated in Fig. 1 (blue), which shows the number of iterations averaged over 10,000 MPC problem instances. For each instance we count the number of iterations required for the algorithm's iterates to satisfy $\|\mathbf{p}_{i+1} - \mathbf{p}_i\|_\infty < \epsilon$ and $\|\mathbf{p}_{i+1} - \mathbf{p}_i\|_\infty < \epsilon\|\mathbf{p}_i\|_\infty$ with $\epsilon = 10^{-3}$. For $N = 2$, problem (10) is infeasible for $\gamma \leq 100$ and $\mathbf{R} = I$, and problem (11) is solved instead. The resuling iteration numbers (Fig. 1, red) show that the FGM requires more iterations for $N = 2$ than for $N = 1$, which is related to the larger constraint set [19]. Fig. 1 confirms that for increasing conditions numbers, the required number of FGM iterations increases.

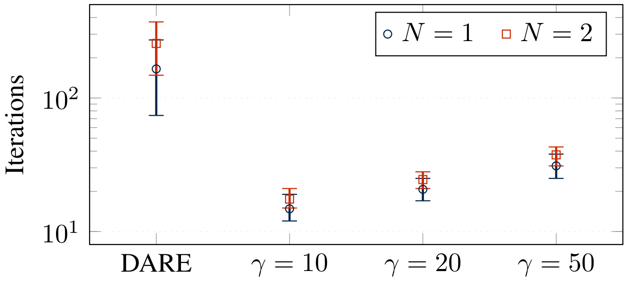

Fig. 1: Average FGM iteration numbers.

The controller performance can be evaluated using the *integrated beam motion* (IBM) [7], which is the cumulative worst-case beam displacement up to a certain frequency, and it is shown in Fig. 2 for $N = 1$ (left) and $N = 2$ (right). For both horizons, the controller performs best when P is obtained from the DARE, and worst when the condition number is limited to $\gamma = 10$. The performance degrades as γ decreases, and the decrease is more pronounced for $N = 2$.

V. Implementation

The Diamond-II controller will be implemented on a VadaTech AMC540 board that embeds an FPGA and two Texas Instruments (TI) C6678 digital signal processors (DSPs) [24] with 8 cores each. For our tests, Alg. 1 is implemented on the DSPs, which are more flexible to program, while the FPGA is responsible for signal routing. The data transfer between FPGA and DSPs takes roughly $5\,\mu\text{s}$. The DSPs are clocked at $1.4\,\text{GHz}$ and the sampling frequency of $10\,\text{kHz}$ therefore allows for 133,000 processor cycles ($95\,\mu\text{s}$). The control problems for the vertical and

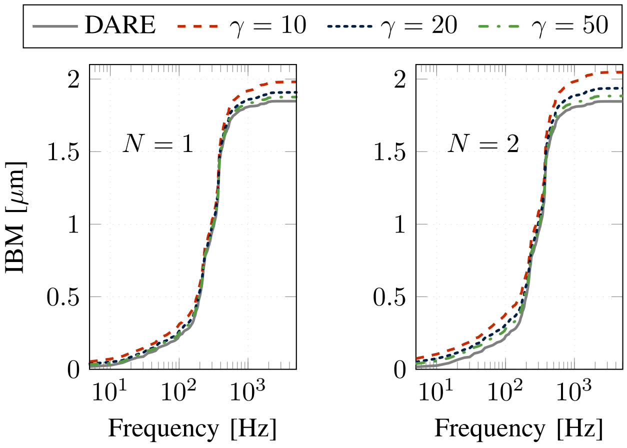

Fig. 2: Integrated beam motion.

horizontal beam directions are independent and one DSP is used for each direction.

The TI C6678 is a floating point processor with two levels of core-local memory (L1/cache, 32 kB and L2, 512 kB) and a third level of shared memory (L3, 4 MB). Accessing the L2 memory is twice as fast as accessing the L3 memory [24].

For the gradient step of Alg. 1, we have implemented an optimized routine that exploits the core architecture and maximizes the cache efficiency. For $N = 1$, the algorithm can be implemented as shown in Alg. 1 and all the problem data, such as the Hessian $\mathbf{J}$, can be saved in L2 memory. For $N = 2$, the Hessian uses almost all of the L2 memory, so some data must be moved to the slower L3 memory. The cache efficiency for the projection can be increased by permuting the data using a perfect shuffle, so that the inputs for magnet i and horizon stages 0 and 1 are contiguous in memory.

If the algorithm is run on a single core with $I_{\max} = 20$, it requires 543 µs for $N = 1$ and 3550 µs for $N = 2$ to compute the control inputs, which is far more than the desired 100 µs. The most expensive operation is the gradient step, which takes 357 µs for $N = 1$ and 3017 µs for $N = 2$. Since the algorithm is dominated by the matrix-vector multiplication of the gradient step, one would expect the computation time to quadruple when doubling the problem size. However, transferring problem data that lies in the L3 memory and cache inefficiencies incur substantial overheads.

The single-core implementation is then parallelized using a standard manager-worker framework, but variable dependencies require core communication and cache operations that are denoted by circled arrows in Alg. 1. For the problem size of the MPC problem (7), the cost of parallelization is not negligible. Fig. 3 shows the overhead introduced by interprocessor communication measured by the elapsed time between a manager request and the acknowledgement of n_w worker cores without worker payload. Three different implementations are compared: The TI Notify scheme, which is a library provided by TI and used by the TI open multiprocessing (openMP) toolbox, the TI multicore navigator (NAV), which is implemented through a separate on-chip processor, and our custom interrupt-free implementation. The TI notification schemes are flexible, but introduce a considerable delay. Note that with 20 synchronization points, the TI Notify scheme alone introduces 200 µs of overhead. For our custom approach, we chose to implement a simpler scheme using integer flags that are saved in the L3 memory. In practice, at each communication step it is also required to invalidate or write-back the cache.

Alg. 1 is sliced into 6 × 32 row-blocks with 192 columns each and deployed on 6 worker cores and 1 manager core. The length of the slices must be a multiple of the cache line size (64 B) and using 7 worker cores would not yield any speed up. The master core coordinates the various steps of Alg. 1 and communicates with the adjacent FPGA. A breakdown of the computation time of Alg. 1 with $I_{\max} = 20$ and $\gamma = 20$ is shown in Fig. 4. For $N = 1$, the algorithm uses 69 µs, which is well below the allowed 100 µs, but for $N = 2$, the computation time of 272 µs is above the time limit. The algorithm would therefore have to be implemented on the FPGA for $N = 2$. However, the very small performance improvement that is obtained when increasing N from 1 to 2 [11] may not justify a complex FPGA implementation.

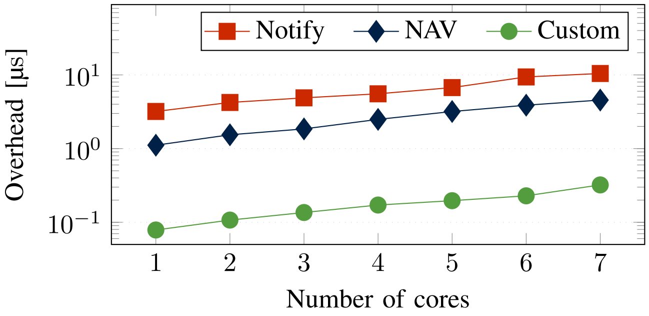

Fig. 3: Interprocessor communication overhead.

Compared to the single-core implementation, the parallelization reduces the computation time by a factor between about 8 and 13. In theory, one would expect the computation time to be reduced by a factor smaller than n_w when deployed onto n_w worker cores. We suspect that this discrepancy is due to memory and cache bandwidth limitations on the single core implementation.

VI. Conclusion

In this paper, we have focused on the practical issues of implementing MPC for the DLS-II electron beam stabilization problem. To obtain an implementation that runs at the desired speed, we tailored the MPC algorithm to the DLS-II system hardware.

Firstly, we avoided removing the time delay by augmenting the system with additional states and designed an observer

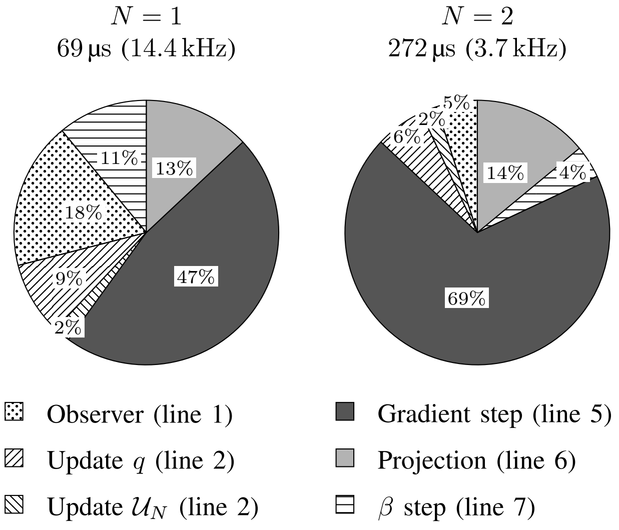

Fig. 4: Computation times for multi-core implementation.

for the delayed states instead. The delayed measurement updates were then projected into the future, which exploited the diagonal structure of the state-space system. Here, it was assumed that all measurements are synchronized. However, at DLS-II, the measurements might have different delays, which could be considered in the observer.

Secondly, because standard preconditioning techniques with diagonal scaling matrices were unable to reduce the condition number of the Hessian, we formulated an optimization problem that finds a stabilizing terminal cost matrix and results in a Hessian with a prescribed condition number. For larger horizons, reducing the condition number of the Hessian also required a modification of the state-weighting matrix. Future research could implement a multi-objective optimization problem that also considers a performance measure of the controller.

Finally, we showed that standard parallelization toolboxes, such as openMP, introduce overheads that would prohibit the algorithm from running at the desired speed, and we therefore implemented a customized core-synchronization framework. Our investigation showed that MPC is applicable to the electron beam stabilization problem, but requires investment of significant effort into the theoretical and practical implementation as well as paying particular attention to details, such as overheads introduced by the CQP initialization or parallelization, which are often neglected in theoretical investigations. Our practical tests also showed that assumptions on computational complexities can be inaccurate, e.g. doubling the CQP problem size does not necessarily result in quadruple computation time nor does parallelizing the algorithm on n_w cores increase the computation speed by a factor of n_w.

In anticipation of our tests, we demonstrated the feasibility for the DLS-I storage ring, but we have not considered a number of additional changes that will be introduced for DLS-II. The number of actuators will be increased from 173 to 396 for Diamond-II, which will significantly increase the computational complexity of the algorithm and further slow down the controller. However, in contrast to the current system, the DLS-II system will have a block-circulant and centrosymmetric symmetry, which can be exploited to increase the computational speed of the controller by a factor of 12 [25]. By exploiting the symmetry and outsourcing initialization routines to the FPGA, the MPC algorithm could be implemented on the DSP for $N \in \{1, 2\}$.

References

[1] C. Abraham et al., "Diamond-II: Conceptual design report," Diamond Light Source Ltd., Didcot, UK, Tech. Rep., May 2019.

[2] W. Heath, "Orthogonal functions for cross-directional control of web forming processes," *Automatica*, vol. 32, no. 2, pp. 183–198, Feb. 1996.

[3] N. Hubert, L. Cassinari, J. C. Denard, A. Nadji, and L. Nadolski, "Global orbit feedback systems down to DC using fast and slow correctors," in *9th Eur. Workshop Beam Diagnostics Instrum. for Part. Accel. (DIPAC)*, Basel, Switzerland, Jan. 2009, pp. 27–31.

[4] C. Schwartz and L. Emery, "Compensating the frequency deadband of the APS real-time and DC transverse orbit correction systems," in *Proc. Part. Accel. Conf. (PAC)*, Chicago, IL, Jun. 2001, pp. 1234–1236.

[5] C. Steier, E. Domming, T. Scarvie, and E. Williams, "Operational experience integrating slow and fast orbit feedbacks at the ALS," in *9th Eur. Particle Accel. Conf.*, Lucerne, Switzerland, 2004.

[6] L. H. Yu, "The performance of a fast closed orbit feedback system with combined fast and slow correctors," in *11th Eur. Particle Accel. Conf.*, Genoa, Italy, 2008.

[7] S. Gayadeen and S. R. Duncan, "Discrete-time anti-windup compensation for synchrotron electron beam controllers with rate constrained actuators," *Automatica*, vol. 67, pp. 224–232, May 2016.

[8] I. Kempf, P. J. Goulart, and S. R. Duncan, "Alternating direction of multipliers method for block circulant model predictive control," in *IEEE 58th Conf. Decis. Control (CDC)*, Nice, France, Dec. 2019, pp. 4311–4316.

[9] K. R. Muske and T. A. Badgwell, "Disturbance modeling for offset-free linear model predictive control," *J. Proc. Control*, vol. 12, no. 5, pp. 617–632, Aug. 2002.

[10] D. Simon, *Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches*, 1st ed. New York, NY, USA: Wiley, 2006.

[11] I. Kempf, P. Goulart, and S. Duncan, "Model predictive control for electron beam stabilization in a synchrotron," March 2021, arXiv:2107.01694 [eess.SY].

[12] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, 1st ed. New York, NY, USA: Wiley, 2005.

[13] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*, 1st ed. Cambridge, UK: Cambridge University Press, 2017.

[14] I. McInerney, E. C. Kerrigan, and G. A. Constantinides, "Horizon-independent preconditioner design for linear predictive control," *IEEE Trans. Automat. Control*, 2022.

[15] X. V. Suo, "Optimal control of an accelerated electron beam using model predictive control," Master's thesis, Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne, Switzerland, Aug. 2017.

[16] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Math. Prog. Comp.*, vol. 12, pp. 637–672, Feb. 2020.

[17] I. Kempf, P. J. Goulart, and S. R. Duncan, "Fast gradient method for model predictive control with input rate and amplitude constraints," in *Proc. 21st IFAC World Congress*, vol. 53, no. 2, Berlin, Germany, July 2020, pp. 6542–6547.

[18] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, 1st ed. Boston, MA, USA: Springer, 2003.

[19] S. Richter, C. N. Jones, and M. Morari, "Computational complexity certification for real-time MPC with input constraints based on the fast gradient method," *IEEE Trans. Automat. Control*, vol. 57, no. 6, pp. 1391–1403, June 2012.

[20] M. K. Tippett, S. E. Cohn, R. Todling, and D. Marchesin, "Conditiioning of the stable, discrete-time Lyapunov operator," *SIAM J. Matrix Anal. Appl.*, vol. 22, no. 1, pp. 56–65, July 2006.

[21] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge, UK: Cambridge University Press, 2013.

[22] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, 1st ed. Philadelphia, PA, USA: SIAM, 1994.

[23] L. El Ghaoui *et al.*, *Advances in Linear Matrix Inequality Methods in Control*, 1st ed. Philadelphia, PA, USA: SIAM, 2000.

[24] *Multicore Fixed and Floating-Point Digital Signal Processor (TMS320C6678)*, Texas Instruments, 2010, sPR5691E.

[25] I. Kempf, P. J. Goulart, S. R. Duncan, and G. Rehm, "Symmetry exploitation in orbit feedback systems of synchrotrons for computational efficiency," *IEEE Trans. Nucl. Sci.*, vol. 68, no. 3, pp. 258–269, March 2021.