Dykstra's Algorithm for the Projection Step in the Model Predictive Control System of a Synchrotron

Claudio Vestini and Idris Kempf

September 2024

1 Context & Background

The forthcoming upgrade of the Diamond Light Source (DLS) synchrotron presents a complex control challenge in the stabilisation of the electron beam. This is due to the increased number of sensors (from 172 to 252) and actuators (from 173 to 396), as well as the introduction of two different types of corrector magnets. Traditional control methods (*modal decomposition* [7]) are no longer sufficient [6], and model predictive control (MPC), while promising, faces high computational demands for high-frequency (100 kHz) control. The optimiser has to solve a constrained quadratic programming (CQP) problem at each timestep in less than $10\mu s$ for feasibility. This involves minimising a cost function to obtain a global minimum, and projecting it onto the constraint set. The projection step can be solved using Dykstra's method [3], which has been shown to converge linearly [8]. This algorithm can stall under certain conditions [1], and this paper will investigate stalling and how it may be prevented.

2 Problem Statement

The relationship between the $n_y = 252$ measured beam displacements $y_k \in \mathbb{R}^{n_y}$ and the $n_u = 396$ actuator inputs $u_k \in \mathbb{R}^{n_u}$ can be described by the state-space representation of a linear system in discrete time,

$$x_{k+1} = Ax_k + Bu_k, \quad u_k \in \mathcal{U}(u_{k-1}), \tag{1a}$$

$$y_k = Cx_k + d_k, \tag{1b}$$

where $x_k \in \mathbb{R}^{n_x}$ are the states at time $t = kT_s$. The inputs u_k are subjected to amplitude and slew-rate constraints that can be modelled as $\mathcal{U}(u_{k-1}) := \mathcal{A} \cap \mathcal{R}(u_{k-1})$, where u_{k-1} is the input applied at time $t = (k-1)T_s$ and $\mathcal{A}$ and $\mathcal{R}(u_{k-1})$

are the amplitude and slew-rate constraint sets:

$$\mathcal{A} \coloneqq \{u_k \in \mathbb{R}^{n_u} \mid -\alpha \le u_k \le \alpha\}, \tag{2a}$$
$$\mathcal{R}(u_{k-1}) \coloneqq \{u_k \in \mathbb{R}^{n_u} \mid -\rho \le u_k - u_{k-1} \le \rho\}. \tag{2b}$$

The condensed [2] model predictive control problem for (1a) is

$$\min_u \frac{1}{2} u^{\mathrm{T}} J u + q(\hat{x}_k, \hat{d}_k)^{\mathrm{T}} u \quad \text{s.t.} \quad u \in \mathcal{S}(u_{-1}), \tag{3}$$

where $u \coloneqq (u_0^{\mathrm{T}}, \dots, u_{N-1}^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^{N n_u}$, N the horizon, $f(u) \coloneqq \frac{1}{2} u^{\mathrm{T}} J u + q(\hat{x}_k, \hat{d}_k)^{\mathrm{T}} u$ the objective function and $J = J^{\mathrm{T}} \in \mathbb{R}^{N n_u \times N n_u}$ the Hessian. The vector $q(\hat{x}_k, \hat{d}_k)$ is an affine function of the observer output $\hat{x}_k$ and $\hat{d}_k$. The closed convex set $\mathcal{S}(u_{-1})$ is defined as

$$\mathcal{S}(u_{-1}) \coloneqq \mathcal{U}(u_{-1}) \times \cdots \times \mathcal{U}(u_{N-2}), \tag{4}$$

and depends on the input u_{-1} applied at time $t-1$. The following assumptions on problem (3) are made throughout the paper:

Assumption I.a (Strong convexity)**.** *It holds that* $0 < \lambda_{min} I \le J \le \lambda_{max} I$, *where* λ_{min} *and* λ_{max} *are the minimum and maximum eigenvalues of* J.

Assumption I.b (Bounded inputs)**.** *The set* $\mathcal{S}$ *is bounded, so that* $|\mathcal{S}| \coloneqq \max_{x,y \in \mathcal{S}} \|x - y\|_2 < \infty$.

To solve problem (3), we are using the fast gradient method shown in Algorithm 1. The present variant of the fast gradient method is applicable to strongly convex functions for ill-conditioned Hessians [9, Ch. 2.2.4]. The fast gradient method requires the Eucledian projection $\mathcal{P}_{\mathcal{S}}$ onto $\mathcal{S}$, and no simple formula exists for $N > 1$. Here, we are replacing the exact projection $\mathcal{P}_{\mathcal{S}}$ with Dykstra's alternating projection algorithm, $\mathcal{D}_{\mathcal{S}}$. Dykstra's method is an iterative algorithm that yields the exact projection if the algorithm is run for an infinite number of iterations. For M iterations, $\mathcal{D}_{\mathcal{S}}(z) \notin \mathcal{S}$ in general[1] and the method yields a projection error that can be quantified as shown in the following assumption.

Assumption II (Approximate projection)**.** *Dykstra's method* $\mathcal{D}_{\mathcal{S}}$ *returns a point* $\mathcal{D}_{\mathcal{S}}(z)$ *that satisfies*

$$\|\mathcal{D}_{\mathcal{S}}(z) - \mathcal{P}_{\mathcal{S}}(z)\|_2 \le \delta(z, M, \mathcal{S}), \tag{5}$$

where the upper bound $\delta(z, M, \mathcal{S}) > 0$ *depends on the maximum number of iterations* M *of Dykstra's method.*

This paper aims to

[1] For example, consider the projection onto a corner (or edge) of two intersecting hyperplanes.

1. Characterize $\delta(z, M, \mathcal{S})$ for $\mathcal{S}$ and $z \in \mathcal{Z}$, where $\mathcal{Z} = \mathcal{Z}(J, q, u_{-1})$ is to be defined.
2. Analyze the converge of Algorithm 1 for fixed M.

The ideal result would be to obtain a formula that relates M and ϵ, where ϵ represents an upper bound on the solution accuracy of the modified fast gradient method.

3 Fast Gradient Method

Algorithm 1 Fast gradient method: Constant step scheme III with parameter [9, Ch. 2.2.4].

Input: $u_{k-1} \rightarrow v_0$
Output: $u_k \leftarrow p_M$

```
1: Set p_0 = 0
2: for i = 0 to M − 1 do
3:     z_{i+1} = (I − J λ_max^{-1}) v_i − q λ_max^{-1}
4:     p_{i+1} = P_S(z_{i+1})
5:     v_{i+1} = (1 + β) p_{i+1} − β p_i
6: end for
```

1: Set $p_0 = 0$
2: **for** $i = 0$ to $M - 1$ **do**
3: $\quad z_{i+1} = (I - J\lambda_{max}^{-1})v_i - q\lambda_{max}^{-1}$
4: $\quad p_{i+1} = \mathcal{P}_{\mathcal{S}}(z_{i+1})$
5: $\quad v_{i+1} = (1+\beta)p_{i+1} - \beta p_i$
6: **end for**

A convergence analysis of Algorithm 2 is presented in [10] with $\mathcal{P}_{\mathcal{S}}$ replaced by $\mathcal{D}_{\mathcal{S}}$ satisfying

$$\|\mathcal{D}_{\mathcal{S}}(z) - \mathcal{P}_{\mathcal{S}}(z)\|_2 \leq \delta, \tag{6}$$

where $\delta > 0$ is an arbitrary constant. Note that Algorithm 1 is obtained from Algorithm 2 by setting $\alpha_0^2 = \lambda_{\max}/\lambda_{\min}$. The authors prove that if

$$\delta \leq \frac{\epsilon\lambda_{\max}^{1/2}}{120\lambda_{\min}^{3/2}\,|\mathcal{S}|}, \tag{7}$$

then at most K iterations of Algorithm 2,

$$K = \left\lceil \sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}} \log\left(\frac{3\lambda_{\min} r_1^2 + 6(f(\mathcal{P}_{\mathcal{S}}(z_1)) - f^\star)}{\epsilon}\right)\right\rceil, \tag{8}$$

are required to obtain a solution p_K that satisfies

$$\operatorname{dist}_{\mathcal{S}}(p_K) \leq \epsilon, \qquad f(p_K) - f^\star \leq \epsilon, \tag{9}$$

where $\operatorname{dist}_{\mathcal{S}}(y) \coloneqq \min_{x\in\mathcal{S}} \|x - y\|_2$, $f^\star = f(u^\star)$ is the minimum of (3) and $r_k \coloneqq \|p_k - u^\star\|_2$.

Algorithm 2 Fast gradient method: Constant step scheme III [9, Ch. 2.2.4] with approximate projection.

Input: $u_{k\text{-}1} \to v_0$
Output: $u_k \leftarrow p_M$

1: Set $p_0 = 0$
2: **for** $i = 0$ to $M - 1$ **do**
3: $z_{i+1} = (I - J\lambda_{max}^{-1})v_i - q\lambda_{max}^{-1}$
4: $p_{i+1} = \mathcal{D}_{\mathcal{S}}(z_{i+1})$
5: $\alpha_{i+1}^2 = (1 - \alpha_{i+1})\alpha_i^2 + \alpha_{i+1}\lambda_{\max}/\lambda_{\min}$
6: $\beta_{i+1} = \alpha_i(1 - \alpha_i)/(\alpha_i^2 + \alpha_{i+1})$
7: $v_{i+1} = (1 + \beta_{i+1})p_{i+1} - \beta_{i+1}p_i$
8: **end for**

4 Dykstra's Alternating Projection

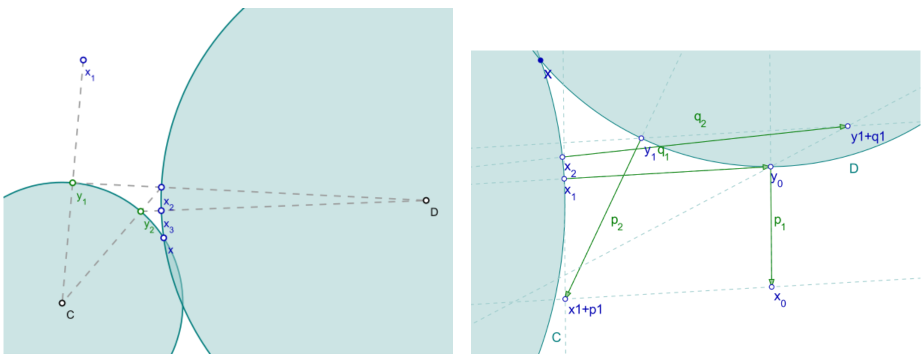
In algorithm 2 the projection step of line 4: is solved using Dykstra's method. Given n convex sets $\mathcal{H}_1, \ldots, \mathcal{H}_n$, Dykstra's alternating projection algorithm [11] finds the orthogonal projection $x^\star$ of x onto $\mathcal{H} \coloneqq \bigcap_{i=1}^{n} \mathcal{H}_i$ by initially setting

$$e_{-n} = e_{-(n-1)} = \ldots = e_{-1} = \vec{0} \tag{10}$$

and generating a series of iterates $\{x_m\}$ using the scheme

$$x_{m+1} = \mathcal{P}_{\mathcal{H}_{[m]}}\left(x_m + e_{m-n}\right), \qquad e_m = x_m + e_{m-n} - x_{m+1}, \tag{11}$$

for $[m] = 0, 1, \ldots$ and $x_0 = x$.
This is a variation of the simpler *Method of Alternating Projections* (MAP), which can be obtained from (11) with the errors $e_m = \vec{0}$. A visual explanation of the difference between the two methods is displayed in Figure 1



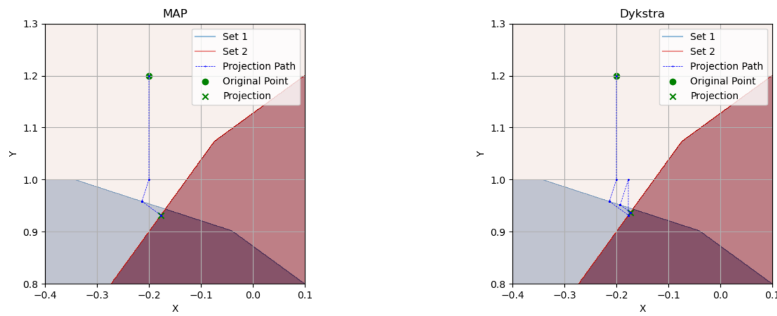
(a) MAP algorithm with successive projections y and x onto sets C and D respectively

(b) Dykstra's algorithm with associated errors p and q

Figure 1: A demonstration of MAP vs Dykstra. Dykstra has intermediate steps

Despite MAP being much simpler, it exhibits weak convergence [4] and does not always produce an optimal output, as illustrated in Figure 2.

The Boyle-Dykstra theorem [3] implies that $\lim_{m\to\infty} \|x_m - \mathcal{P}_{\mathcal{H}}(x)\| = \vec{0}$. For a finite number of iterations, there is no guarantee that $x_m \in \mathcal{H}$.



(a) Sub-optimal solution for MAP after 1 iteration. Nothing happens in second iteration

(b) Dykstra gets closer to optimal solution in second iteration

Figure 2: A demonstration of solution optimality. Both algorithms were ran for $m = 2$ iterations. Dykstra's will produce an optimal output after infinitely many iterations

Here, we assume that $\mathcal{H}$ is a polyhedron and the $\mathcal{H}_i$ are halfspaces given by

$$\mathcal{H}_i := \{ x \mid \langle x, f_i \rangle \leq c_i \}, \tag{12}$$

where $\|f_i\| = 1$. In addition, define boundaries

$$H_i := \{ x \mid \langle x, f_i \rangle = c_i \}, \tag{13}$$

so that $\operatorname{int} \mathcal{H}_i := \mathcal{H}_i \backslash H_i$. The projections onto H_i and $\mathcal{H}_i$ are given by

$$\mathcal{P}_{H_i}(x) = x - (\langle x, f_i \rangle - c_i) f_i, \qquad \mathcal{P}_{\mathcal{H}_i}(x) = \begin{cases} x & \text{if } x \in \mathcal{H}_i \\ \mathcal{P}_{H_i}(x) & \text{if } x \notin \mathcal{H}_i \, . \end{cases} \tag{14}$$

For this particular choice of sets, $e_m = k_m f_{[m]}$ with $k_m = \operatorname{dist}_{\mathcal{H}_{[m]}}(x_{m-1} + e_{m-n})$, i.e. the auxiliary vector is either 0 or parallel to $f_{[m]}$. The convergence of Dykstra's method has been analyzed in [8, 5], where it has been shown that the convergence is linear. In [5], the proof is based on partitioning the set $\{1, \ldots, n\}$ into

$$A = \{ i \in \{1, \ldots, n\} \mid x_\infty \in H_i \}, \quad B = \{1, \ldots, n\} \backslash A = \{ i \in \{1, \ldots, n\} \mid x_\infty \in \operatorname{int} \mathcal{H}_i \}, \tag{15}$$

where $x_\infty = \lim_{m \to \infty} x_m$. It can be shown that there exists a number N_1 such that whenever

$$[m] \in B, \quad m \geq N_1 \quad \Rightarrow \quad x_m = x_{m-1}, \quad e_m = \vec{0}, \tag{16}$$

i.e. the half-spaces in B become "inactive". Furthermore, there exists $N_2 \geq N_1$ such that whenever $n \geq N_2$, it holds that

$$\|x_{m+n} - x_\infty\|_2 \leq \alpha_{[m]} \|x_m - x_\infty\|_2, \tag{17}$$

where $0 \leq \alpha_{[m]} < 1$. With these ingredients, the following result is obtained:

Theorem 1. *There exist constants* $0 \leq c < 1$ *and* $\rho > 0$ *such that*

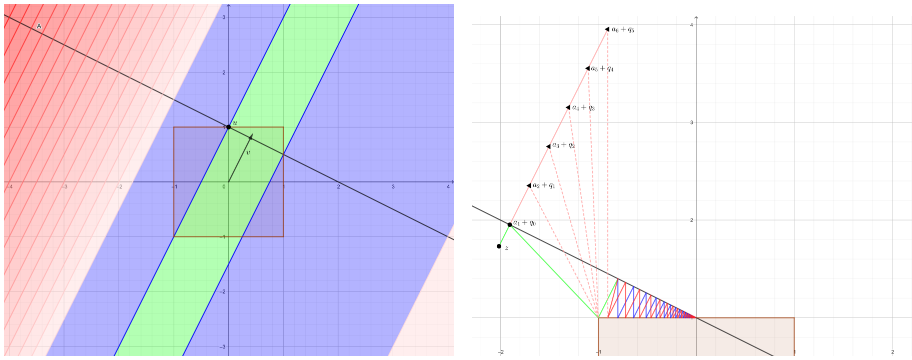
$$\|x_m - x_\infty\| \leq \rho c^m.$$

The factor c can be estimated from the smallest $\alpha_{[m]}$, which is characterized by the angle between certain subspaces (subspaces formed by the "active" halfspaces). The factor $\alpha_{[m]}$ can be upper-bounded by considering the "worst" angles in the polyhedron. The constant ρ, however, depends on a number $N_3 \geq N_2$ and on x. It is unclear how to obtain that constant ρ. In fact, the authors of [11] and [12] emphasize that ρ cannot be computed in advance, and that the inability to compute a bound on the projection error makes the application of Dykstra's method difficult. The authors of [11] proposed a combined Dysktra-conjugate-gradient method that allows for computing an upper bound on $\|x_m - x_\infty\|$. The authors of [12] proposed an alternative algorithm called *successive approximate algorithm*, which promises fast convergence, conditioned on knowing a point $x \in \mathcal{H}$ in advance.

5 Stalling

In [1], the behaviour of Dykstra's method is analysed for two sets. The authors give conditions on Dykstra's algorithm for (1) finite convergence, (2) infinite convergence and (3) stalling followed by infinite convergence. A specific example is given for the case that the set is provided by the intersection of a line with a unit box in $\mathbb{R}^2$ ($\mathcal{H}$ is a polyhedron). It can be shown that cases (1)–(3) depend on the starting point x_0, and one can determine the 3 regions shown in Figure 3a that yield different convergence behaviour. Convergence case 1 is obtained when starting in the green region, case 2 when starting in the blue region and case 3 when starting in the red region. To understand the stalling effect, consider Figure 3b, which shows the first iterations of Dykstra's algorithm with starting point in the red region. Note that the outcome of Dykstra's algorithm depends on the order of the sets $\mathcal{H}_i, \ldots, \mathcal{H}_n$. In Figure 3b, the algorithm starts by projecting onto the box and then onto the line. It can be seen that for the first 6 iterations[2], Dykstra's algorithm returns the top left corner of the box ("stalling"). The authors also determine the exact number of iterations required to break free from the red region, and show that if the starting point is arbitrarily far to the left, the algorithm will need an arbitrarily large iteration number to break free from the red region. It seems intuitive from the mechanics of Dykstra's algorithm that the reason why this case stalls is the 'pointy' nature of the box's set. The algorithm keeps projecting onto the corner from a number of points, until it can project onto the top side of the box, at which point it exits stalling.

[2]By one iteration we mean one cycle of n projections here.

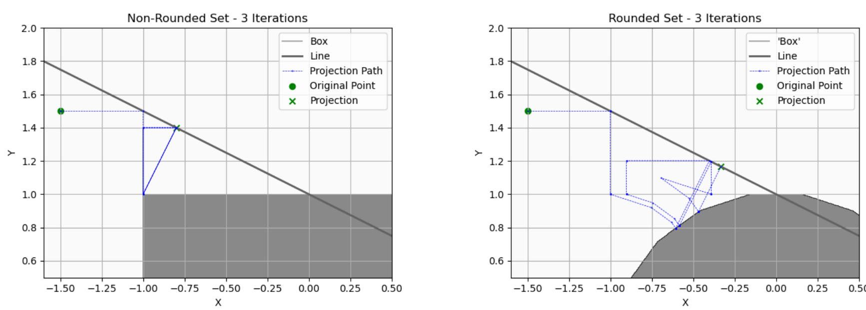


(a) Line-box example with different regions that yield different convergence properties.

(b) Stalling for the line-box example when x_0 is in the red region.

Figure 3: A demonstration of the stalling problem for a box and a line

If the boundary of the box was 'smooth', there would simply be no corner to get stuck at. The algorithm would get closer to the optimal solution at every step. This is illustrated in Figure 4.



(a) Dykstra's algorithm on the box and line example

(b) Rounded edges prevent algorithm from stalling

Figure 4: A demonstration of how roundedness prevents stalling

6 Thoughts

In theory, the formulae presented in these notes would allow for determining the solution accuracy of the fast gradient method if both the fast gradient method and Dykstra's algorithm are run for a fixed number of iterations. However, because the error bound for Dykstra's method cannot easily be computed in advance[3], the formulae cannot be applied.

The results on the fast gradient method are surprising. I do not understand how one can obtain a solution with ϵ accuracy if the approximate projection returns the unchanged vector, for example. However, it does seem that the proof requires the projection error to be bounded by δ (regardless of the starting point), in which case the former example does not work.

It seems obvious that there is a connection between the stalling and the inability to compute the error bound in advance. E.g. consider the stalling situation from last section (Figure 3b). It is my impression that the constant c in the Dykstra error bound could be computed after the stalling period (note that N_2 is reached after the stalling period – one needs to model the box via halfplanes, in which case we enter the interior of the halfplane defining the left side of the box).

Questions:

- Can Dysktra's method be modified to escape the stalling region in a finite number of iterations?
  It does seem that introducing $e_m = e_{m-n} + \beta_m(x_{m-1} - x_m)$ improves things.
- Can Dysktra's method be accelerated after escaping the stalling region?
- Can we replace Dykstra's method by the *successive approximate algorithm* [12]?
  The previously computed input is always feasible.
- Can we modify Dykstra's method such that it becomes independent of the set ordering?
  This should be possible to formulate mathematically. However, it does make sense to obtain different "trajectories". The convergence properties should be independent of set ordering.
- Is there some heuristic for the set ordering?
  I have tried a few without success.

7 Acceleration for Dykstra's Method

Consider introducing the step size parameter $\beta_m \geq 0$:

$$x_m = \mathcal{P}_{\mathcal{H}_{[m]}}(x_{m-1} + e_{m-n}), \qquad e_m = e_{m-n} + \beta_m(x_{m-1} - x_m). \tag{18}$$

For $\beta_m = 0$, we obtain Von Neumann's *Alternating Projection Method* and for $\beta_m = 1$, we obtain Dykstra's method (11). We proceed by characterising the term e_m.

[3] For an arbitrary polyhedral set. It can be computed for e.g. 2 sets.

Lemma 1. *It holds that $e_m = y_m f_{[m]}$, where*

$$y_n = (1 - \beta_m) y_{m-n} + k_m,$$

and $k_m = dist_{\mathcal{H}_{[m]}}(x_{m-1} + e_{m-r})$.

Proof. Suppose that $x_{m-1} + e_{m-n} \in \operatorname{int} \mathcal{H}_{[m]}$. Then, $x_n = x_{m-1} + e_{m-n}$ and

$$e_m = (1 - \beta_m) e_{m-n}.$$

Suppose that $x_{m-1} + e_{m-n} \notin \operatorname{int} \mathcal{H}_{[m]}$. Then, $x_m = x_{m-1} + e_{m-n} - \left(\left\langle x_{m-1} + e_{m-n}, f_{[m]} \right\rangle - c_i \right) f_{[m]}$ and

$$e_m = (1 - \beta_m) e_{m-n} + k_m f_{[m]}.$$

Note that $e_m = 0$ for $m \leq 0$. By induction, e_m is always parallel to $f_{[m]}$ or zero. Substituting $e_m = y_n f_{[m]}$ yields

$$y_m = (1 - \beta_m) y_{m-n} + k_m.$$

□

References

[1] H. H. Bauschke, R. S. Burachik, D. B. Herman, and C. Y. Kaya. On Dykstra's algorithm: finite convergence, stalling, and the method of alternating projections. *ArXiv e-prints*, pages 1–14, January 2020.

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

[3] J. P. Boyle and R. L. Dykstra. A method for finding projections onto the intersection of convex sets in hilbert spaces. In *Advances in Order Restricted Statistical Inference*, pages 28–47. Springer New York, 1986.

[4] L.M. Bregman. The method of projections for finding the common point of convex sets. *Sov. Math. Dokl.*, 6:688–692, 1965.

[5] F. Deutsch and H. Hundal. The rate of convergence of Dykstra's cyclic projections algorithm: The polyhedral case. *Numerical Functional Analysis and Optimization*, 15(6):537–565, May 1994.

[6] Idris Kempf et al. Model predictive control for electron beam stabilization in a synchrotron. *IFAC*, Jul. 2020.

[7] W.P. Heath. Orthogonal functions for cross-directional control of web forming processes. *Automatica*, 32(2):183–198, February 1996.

[8] A. N. Iusem and A. R. de Pierro. On the convergence properties of Hildreth's quadratic programming algorithm. *Math. Prog. Ser. A and B*, 47(1):37–51, May 1990.

[9] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Springer Science & Business Media, 2003.

[10] A. Patrascu and I. Necoara. On the convergence of inexact projection primal first-order methods for convex minimization. *IEEE Trans. on Automat. Cont.*, 63(10):3317–3329, Oct. 2018.

[11] C. Perkins. A convergence analysis of Dykstra's algorithm for polyhedral sets. *SIAM J. Numer. Anal.*, 40(2):732–804, July 2002.

[12] S. Xu. Successive approximate algorithm for best approximation from a polyhedron. *J. Approx. Theory*, 93(3):415–433, June 1998.