Splitting methods in control

Giorgos Stathopoulos[1], Alexander Szücs[1], Ye Pu[1] and Colin N. Jones[1]

Abstract— **The need for optimal control of processes under a restricted amount of resources renders first order optimization methods a viable option. Although computationally cheap, these methods typically suffer from slow convergence rates. In this work we discuss the family of first order methods known as *decomposition schemes*. We present three popular methods from this family, draw the connections between them and report all existing results that enable acceleration in terms of the convergence rate. The approach for splitting a problem into simpler ones so that the accelerated variants can be applied is also discussed and demonstrated via an example.**

I. INTRODUCTION

The significant progress that has been made in recent years both in hardware implementations and in numerical computing has rendered real-time optimization-based control a viable option when it comes to advanced industrial applications. More recently, the need for control of a process in the presence of a limited amout of hardware resources has triggered research in the direction of embedded optimization-based control. Many efficient high-speed solvers have been developed for both linear and nonlinear control, based on either *first order methods* (FiOrdOs [1]), *interior point (IP) methods* (FORCES [2], CVXGEN [3]) or active sets (QPOASES [4]).

In this work we focus on systems with linear dynamics, giving rise to convex control problems. The purpose of the survey is to explore a family of first order methods known as *decomposition schemes* or *operator splitting methods*. In the simplest case, the abstract form of the problem at hand is the minimization of the sum of two convex functions and can be written as

$$\text{minimize} \quad f(x) + g(Ax) \ , \tag{1}$$

with variables $x \in \mathbb{R}^n$, where $f : \mathbb{R}^n \to (-\infty, \infty]$ and $g : \mathbb{R}^m \to (-\infty, \infty]$ are proper, lower semi-continuous (lsc) convex functions and $A : \mathbb{R}^n \to \mathbb{R}^m$ is a linear map. A splitting method can be applied to the above problem after rewriting it as

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax = z \ , \end{array} \tag{2}$$

by alternatively (or simultaneously) minimizing over f and g. A dual variable update for the equality constraint ensures that the solutions of problems (2) and (1) are identical. Inequality constraints are already present in the formulation in the form of indicator functions, *i.e.*, a membership function for a set C

$$\delta_C(x) = \begin{cases} 0 & x \in C \\ \infty & \text{otherwise.} \end{cases} \tag{3}$$

Formulations similar to the above have been studied extensively and we can look for their roots in the method of multipliers [5], the Arrow-Hurwicz method [6], Douglas-Rachford splitting [7], and ADMM [8], [9]. More recent references that illustrate the applicability of such methods in modern engineering problems (signal and image processing, big data analysis, machine learning) are [10] and [11]. The thesis [12] provides a nice and comprehensive description of the connection of several splitting algorithms under a common framework. Finally, the book [13] provides a mathematically rigorous introduction to operator splitting methods in general Hilbert spaces.

Although it is established that splitting methods are quite beneficial when applied to large-scale problems, their potential in solving small to medium scale embedded optimization problems has not been studied so extensively. Our purpose is to study the behavior of such algorithms as solvers of control-related problems of that scale. Our effort focuses on identifying special characteristics of these problems and how they can be exploited by some popular splitting methods. Some of the questions that we attempt to answer are:

1) It is very common in practice that optimal control problems come with a quadratic objective, since in this way stability can be proven for regulation or tracking purposes. What is the best way to exploit this smooth term?
2) Given that a control problem has to be solved repeatedly (*e.g.*, MPC), how can warm-starting affect the speed?
3) Given the structure of the problem at hand, which algorithms will converge more quickly?

In what follows we present three popular splitting algorithms, the *Alternating direction method of multipliers (ADMM)*, the *Alternating minimization algorithm (AMA)* and the *primal-dual scheme from Chambolle and Pock (CP)*. Our choice is motivated from the fact that the methods are, analyzed and extended from several communities, and their properties are well-understood.

The paper is organized as follows: In Section II we formulate the problem we want to solve and look at it from three different perspectives, resulting in the three algorithms we use. Subsequently we introduce the algorithms under a unified scheme and report their properties, as well as how they can be applied to our problem formulation. In Section III we explain how one can exploit the structure of the problem to accelerate the theoretical convergence rates. An example in Section IV demonstrates the behavior of the three algorithms.

[1] The authors are with Laboratoire d'Automatique, EPFL, CH-1015 Lausanne, Switzerland
{georgios.stathopoulos,alexander.szucs,y.pu,colin.jones}@epfl.ch

II. THE ALGORITHMS

A. The different representations

We narrow the general formulation to our problems of interest which can, without loss of generality, be written as

$$\begin{array}{ll} \text{minimize} & (1/2)z^T Qz + c^T z + \sum_{i=1}^{M} l_i(T_i z + t_i) \\ \text{subject to} & Az = b \ , \end{array} \tag{P}$$

with variable $z \in \mathbb{R}^n$, where $Q \in \mathbb{S}^n_+$, and $T_i \in \mathbb{R}^{p_i \times n}$. The following assumption holds:

Assumption 1: The functions $l_i : \mathbb{R}^{p_i} \rightarrow (-\infty, \infty]$ are closed, lsc convex functions.

Formulation (P) is quite general and can describe any convex optimization problem. The choice of the quadratic part $(1/2)z^T Qz + c^T z$ and the equality constraints $Az = b$ being represented in an explicit way is motivated by the standard form that control problems take.

For lighter notation, we define $f(z) := \left\{(1/2)z^T Qz + c^T z \mid Az = b\right\}$. We also denote the concatenated vectors and matrices associated with the affine term in the l_i's as $T = (T_1, \ldots, T_M)$ and $t = (t_1, \ldots, t_M)$. Using slack variables $y_i = T_i z + t_i$, $i = 1, \ldots, M$, the *Lagrangian* for (P) is written as

$$L = f(z) + \sum_{i=1}^{M} l_i(y_i) + \sum_{i=1}^{M} \langle \lambda_i, -t_i - T_i z + y_i \rangle \ , \tag{L}$$

where $\lambda_i \in \mathbb{R}^{p_i}$ are dual variables associated with the equality constraints introduced above. We can recover the optimum by solving

$$(\lambda^\star, z^\star, y^\star) = \underset{\lambda}{\operatorname{argmax}}\, \underset{z,y}{\operatorname{argmin}}\, L(\lambda, z, y) \ , \tag{4}$$

where $\lambda = (\lambda_1, \ldots, \lambda_M) \in \mathbb{R}^p$, $y = (y_1, \ldots, y_M) \in \mathbb{R}^p$, $p = \sum_{i=1}^{M} p_i$. For solving problem (P) we consider three approaches, namely solving a saddle point problem either on the Lagrangian, the augmented Lagrangian function or a generic saddle-point formulation that involves taking the Legendre-Fenchel dual of the functions $l_i(\cdot)$.

The *augmented Lagrangian* for problem (P) is defined by

$$\begin{aligned} L_\rho = & f(z) + \sum_{i=1}^{M} l_i(y_i) + \sum_{i=1}^{M} \langle \lambda_i, -t_i - T_i z + y_i \rangle + \\ & \frac{\rho}{2} \sum_{i=1}^{M} \| -t_i - T_i z + y_i \|^2 \ , \end{aligned} \tag{AL}$$

for $\rho > 0$ and the problem to solve becomes

$$(\lambda^\star, z^\star, y^\star) = \underset{\lambda}{\operatorname{argmax}}\, \underset{z,y}{\operatorname{argmin}}\, L_\rho(\lambda, z, y) \ . \tag{5}$$

Another option is to apply some partial dualization to the Lagrangian formulation, resulting in a primal-dual equivalent that is easier to solve. Making use of the Legendre-Fenchel conjugate,

$$l_i^\star(\lambda_i) = \sup_{z} \langle T_i z + t_i, \lambda_i \rangle - l_i(T_i z + t_i) \ ,$$

the functions $l_i(T_i z + t_i)$ can now be expressed as

$$l_i(T_i z + t_i) = \sup_{\lambda_i} \langle T_i z + t_i, \lambda_i \rangle - l_i^\star(\lambda_i) \ .$$

In this way the affine argument of $l_i(\cdot)$ appears in a bilinear term and $l_i^\star(\cdot)$ becomes a function of a simple argument. Consequently we can solve the saddle-point formulation

$$(z^\star, \lambda^\star, \nu^\star) = \underset{z \in Z}{\operatorname{argmin}}\, \underset{\lambda, \nu}{\operatorname{argmax}}\, S(z, \lambda, \nu) \ , \tag{6}$$

where

$$S = \langle Tz + t, \lambda \rangle + \langle Az - b, \nu \rangle + (1/2)z^T Qz + c^T z - \sum_{i=1}^{M} l_i^\star(\lambda_i) \ . \tag{S}$$

Note that the equality constraints $Az = b$ are now treated explicitely by means of the multiplier ν. It is interesting that for indicator functions of convex cones, the Legendre-Fenchel dual is the indicator function of the polar cone, rendering the evaluation of $l_i^\star$ easy, especially for the standard self-dual cones.

B. A unified framework

The three approaches for solving (P), *i.e.*, (4),(5) and (6) originate from Rockafellar's *Proximal method of multipliers* [14]. When applying decomposition to this method, we obtain a unified framework for the three algorithms, known as the *Proximal alternating direction method of multipliers (PADMM)* which is written as:

Algorithm 1 Proximal Alternating Direction Method of Multipliers (PADMM)

```
Require: Initialize $z^0 \in \mathbb{R}^n$, $y_i^0 \in \mathbb{R}^{p_i}$, $\lambda^0 \in \mathbb{R}^{p_i}$, and $\rho > 0$
  loop
    1: $z^{k+1} = \underset{z}{\operatorname{argmin}} \quad f(z) + \sum_{i=1}^{M} \left\langle \lambda_i^k, -T_i z \right\rangle +$
           $(\rho/2) \sum_{i=1}^{M} \| -t_i - T_i z + y_i^k \|^2 + (1/2) \| z - z^k \|_{P_1}^2$
    2: $y_i^{k+1} = \underset{y_i}{\operatorname{argmin}} \quad l_i(y_i) + \left\langle \lambda_i^k, y_i \right\rangle + (\rho/2) \| -t_i -$
           $T_i z^{k+1} + y_i \|^2 + (1/2) \| y_i - y_i^k \|_{P_{2i}}^2$, $i = 1, \ldots, M$
    3: $\lambda_i^{k+1} = \lambda_i^k + \rho(-t_i - T_i z^{k+1} + y_i^{k+1})$, $i = 1, \ldots, M$
  end loop
```

Algorithm 1 comes with many names, *e.g.*, *Linearized proximal method of multipliers (L-PMM)* [15], *Split Inexact Uzawa (SIU)* [16], *Generalized Alternating Direction Method of Multipliers (GADMM)* [17]. The matrices P_1, P_{2i} are positive semidefinite and offer some flexibility in preconditioning the proximal term. The second step of the algorithm is a *proximal minimization step* and can be written via the *prox* operator of a function, defined as

$$\mathbf{prox}_{\rho f}(x) := \inf_{y \in Y} \left\{ f(y) + \frac{1}{2\rho} \|y - x\|^2 \right\} \ .$$

From this scheme we can recover:

- Alternating direction method of multiplier (ADMM) [8], [9]: We set $P_1 = 0$ and $P_{2i} = 0$. ADMM converges in function values $f(z^k) + \sum_{i=1}^{M} l_i(y_i^k) \rightarrow p^\star$, in the residual $y^k - Tz^k - t \rightarrow 0$, as well as to the dual optimum $\lambda^\star$

for an arbitrarily large stepsize ρ and with no extra assumptions.

Algorithm 2 Alternating direction method of multiplier (ADMM)

```
Require: Initialize $z^0 \in \mathbb{R}^p$, $\lambda^0 \in \mathbb{R}^p$, and $\rho > 0$
  loop
    1: $z^{k+1} = \operatorname*{argmin}_z \quad f(z) + \sum_{i=1}^{M} \left\langle \lambda_i^k, -T_i z \right\rangle +$
           $(\rho/2) \sum_{i=1}^{M} \| -t_i - T_i z + y_i^k \|^2$
    2: $y_i^{k+1} = \mathbf{prox}_{\frac{1}{\rho} l_i} \left( T_i z^{k+1} + t_i - \lambda_i^k / \rho \right),\ i = 1, \dots, M$
    3: $\lambda_i^{k+1} = \lambda_i^k + \rho(-t_i - T_i z^{k+1} + y_i^{k+1}),\ i = 1, \dots, M$
  end loop
```

- Alternating minimization algorithm (AMA) [18]: The algorithm is a hybrid scheme, consisting of minimizing the original Lagrangian (L) in Step 2, and the augmented one (AL) in Step 3 (drop all colored terms in Algorithm 1). In this way, the quadratic coupling that comes from the augmented Lagrangian term in the first step vanishes, allowing for further decomposition if the structure of f permits to do so. In order to ensure convergence, the stepsize ρ has to be taken as $\varepsilon \leq \rho \leq \frac{4\sigma_f}{\|T\|^2} - \varepsilon$, where $\varepsilon \in (0, \frac{2\sigma_f}{\|T\|^2})$ and f has to be strongly convex, with convexity modulus σ_f. Under these assumptions, convergence of the primal sequence $z^k \to z^\star$, the dual sequence $\lambda^k \to \lambda^\star$ and the residual sequence $y^k - Tz^k - t \to 0$ can be proven [18].

Algorithm 3 Alternating minimization algorithm (AMA)

```
Require: Initialize $\lambda^0 \in \mathbb{R}^p$, and $\rho$ within permitted range
  loop
    1: $z^{k+1} = \operatorname*{argmin}_z \quad f(z) + \sum_{i=1}^{M} \langle \lambda_i^k, -T_i z \rangle$
    2: $y_i^{k+1} = \mathbf{prox}_{\frac{1}{\rho} l_i} \left( T_i z^{k+1} + t_i - \lambda_i^k / \rho \right),\ i = 1, \dots, M$
    3: $\lambda_i^{k+1} = \lambda_i^k + \rho(-t_i - T_i z^{k+1} + y_i^{k+1}),\ i = 1, \dots, M$
  end loop
```

- Chambolle-Pock primal-dual scheme, basic version (CPI) [19]: Chambolle and Pock's scheme solves problem (6) by means of the alternation procedure (presented in Algorithm 4) which is seemingly different from Algorithm 1.

Algorithm 4 Chambolle-Pock I (CPI)

```
Require: Initialize $\lambda^0 \in \mathbb{R}^p$, $\nu^0 \in \mathbb{R}^m$ $z^0 \in \mathbb{R}^n$. Choose $\tau, \rho >$
    $0$ and $\tau\rho\|(T,\ A)\|^2 < 1,\ \theta \in [0,1]$.
  loop
    1: $\lambda_i^{k+1} = \mathbf{prox}_{\rho l_i^\star} \left( \lambda_i^k + \rho(T_i \bar{z}^{k+1} + t_i) \right),\ i = 1, \dots, M$
    2: $\nu^{k+1} = \nu^k + \rho(A\bar{z}^k - b)$
    3: $z^{k+1} = \operatorname*{argmin}_{z \in Z} \quad (1/2) z^T Q z + c^T z + \sum_{i=1}^{M} T_i^T \left\langle z, \lambda_i^{k+1} \right\rangle +$
           $\left\langle z, A^T \nu^{k+1} \right\rangle + (1/2\tau)\|z - z^k\|^2$
    4: $\bar{z}^{k+1} = z^{k+1} + \theta(z^{k+1} - z^k)$
  end loop
```

As is proven in [15], Algorithm 4 is equivalent to Algorithm 1, for the special choices $P_{2i} = 0$ and $P_1 = (1/\tau)I - \rho \sum_{i=1}^{M} T_i^T T_i$, with $\theta = 1$. In this way, Algorithm 4 linearizes the quadratic term that appears in Step 1 of Algorithm 1 and hence decouples the minimization problem. Note that AMA achieves the same decoupling, but in a different way. The cost of simplifying the optimization problem comes, as in AMA, with restrictions to the stepsizes, since the condition $\tau\rho\|(T,\ A)\|^2 < 1$ has to hold.

III. ACCELERATED CONVERGENCE

There are various extensions of the three methods we presented that can significantly improve their performance in practical applications. In general there are two ways to improve timings:

1) Improving the theoretical convergence rates, which is done by exploiting properties of the functions in (P).
2) Speeding up the computations, which can be done is several ways, *e.g.*, fast numerical linear algebra, preconditioning of the data.

In many cases the two approaches are competing. For example, one can precondition the problem so that an accelerated variant of a method can be used, but at the same time some favorable sparsity pattern of the original problem is lost. In our experience, there is no 'golden rule' when it comes to choosing a particular method and applying the various extensions for speeding it up. The choice of the method should be motivated from the problem's structure and vice-versa. In the subsections that follow we aim at providing the reader with a wide overview of several variants of the methods that improve the convergence rates. Computational speedup is not explored in the current version of the article due to space limitations.

A. How to split

The first question that comes to mind when using a splitting method is how to perform the splitting. This choice can heavily affect the speed of the algorithm. Choosing a splitting pattern is equivalent to formulating the two subproblems that have to be solved in the algorithmic schemes 2, 3 or 4. Consequently, the choice will also restrict the options for acceleration. A general guideline would be the following:

1) Both subproblems should have a closed form solution if possible; if not, they should be cheap to solve. The whole purpose of using splitting on (P) is to end up with simpler subproblems.
2) More precisely, the proximal step should be simple to solve. The step constitutes often of projections onto simple constraint sets, or proximal minimizations with respect to norms.
3) Expensive operations, like matrix inversions, should be avoided. If there are quantities that do not change during the execution, they should be prefactored.
4) If an accelerated version of an algorithm can be used without heavily altering a well-structured problem, then it should be used.

ADMM: In this case, most of the flexibility comes in Step 2, since Step 3 is either a simple projection or a proximal minimization operation, provided l_i is simple. The augmented Lagrangian term will contribute with a quadratic term of the form $(\rho/2)z^T\left(\sum_{i=1}^{M} T_i^T T_i\right)z$ to the objective, hence even if Q is a diagonal matrix, the resulting quadratic term is most probably dense. In this sense, one can either minimize the resulting quadratic function restricted to the subspace $Az = b$, *i.e.*, solve a KKT system (see [20]), or by eliminating the equality constraint. Note that this is equivalent to taking a Newton step on a quadratic perturbation of $f(z)$, which explains why this approach needs relatively few iterations for convergence. The bottleneck is the matrix inversion that has to be performed at each iteration. If ρ is constant, one can use either a sparse *LDL* factorization on the KKT system, or a Cholesky factorization in the second case and consequently solve by means of forward-backward substitution [21, Appendix C].

AMA: The method is applicable under the assumption that f is strongly convex. On the other hand, if the assumption holds and f has some structure (*e.g.*, diagonal, block diagonal), the method should be preferred since the matrix inversion can be very cheap. In several MPC applications this is not the case though, since, in order to ensure strong convexity, f becomes a dense quadratic form for the condensed problem. Note that the spectral radius of T and the minimum eigenvalue of the quadratic term will affect the choice of the stepsize, many times leading to a very small one.

CPI: This method combines properties of the other two, in the sense that the first step is still decoupled but there is no strong convexity assumption. In order to avoid densification of the quadratic term, we choose to treat the equality constraints in a Lagrangian fashion (Step 2), a choice that, along with the stepsizes' limitations, can render the algorithm slow to converge in iterations' number. Keeping Step 3 simple allows for moving some (simple) constraints directly in the objective ($z \in Z$), if the resulting optimization problem has a closed form solution. The algorithm is built such that it favors simple computations in the expense of more iterations.

B. Improvements in the convergence rate

All three schemes have benefited from *Nesterov's optimal relaxation sequence* as introduced in [22]. Nesterov's method is a variant of gradient descent, where, instead of a gradient descent update $\{x^k\}$ sequence one uses the over-relaxed sequence $\{\hat{x}^k\}$:

$$\alpha^{k+1} = \left(1+\sqrt{4(\alpha^k)^2+1}\right)/2$$

$$\hat{x}^{k+1} = x^k + \frac{\alpha^k - 1}{\alpha^{k+1}}(x^k - x^{k-1}) \ , \tag{7}$$

with $\alpha^0 = 1$. Application of the scheme results in an $O(1/k^2)$ global rate of convergence in function values; a rate that is optimal for first order methods. Convergence in terms of the sequences is trickier to prove. Roughly speaking, when the optimal $O(1/k^2)$ rate in terms of the primal (dual) function values is achieved, the primal (dual) sequences converge with rate $O(1/k)$ [23], [15], [19].

Linear convergence rates have also been proven for ADMM and CP methods under specific assumptions on the structure of problem (P). Due to space limitations we only present the extensions of the methods that are based on Nesterov's acceleration or similar techniques, and we collect all other special cases in a table in the end of the section.

ADMM: For ADMM, convergence of the sequences $\{z^k\},\{y^k\},\{\lambda^k\}$ with rate $O(1/\sqrt{k})$ is proven in the recent work [15]. These rates are *global* and come *with no further assumptions on the structure of the problem*.

A fast version of the method (FADMM), based on Nesterov's acceleration, was first presented in [23]. Nesterov's optimal relaxation is applied on the sequences $\{y^k\}$ and $\{\lambda^k\}$. The authors use an *adaptive restarting scheme* [24] based on the residuals' error (see Appendix A). Since the accelerated sequences often exhibit an oscillatory behavior and might over(under)shoot the optimal value, a check is performed, and if the residuals increase in two subsequent iterations, the acceleration scheme is reset.

FADMM can be shown to have a global $O(1/k^2)$ convergence rate in the dual function's values under the assumption that f and l_i are strongly convex and furthermore l_i are quadratic. In the absence of these limiting assumptions, we can have an empirically fast convergence with unproved rate. All details are given in [23]. Note that FADMM can be applied to the same family of problems as ADMM with no extra assumptions and small additional computational cost.

FAMA: The accelerated version of AMA makes use of Nesterov's acceleration scheme on the dual sequence $\{\lambda^k\}$ [23]. Under the same stepsize restriction as in the basic version, *convergence of the dual objective value* at rate $O(1/k^2)$ has been proven, inspired from the convergence proof of the FISTA algorithm [25]. Same as with FADMM, FAMA can practically be applied to every problem that AMA can solve.

CPII: For the basic version of CP (CPI), a partial primal-dual gap is shown to shrink with rate $O(1/k)$ in an ergodic sense for the sequences $\{z^k\}$, $\{\lambda^k\}$ and $\{v^k\}$ in [19]. CP algorithm comes with an accelerated variant, under the assumption that f is uniformly convex, denoted here as *the second method* of Chambolle and Pock (CPII). The acceleration is achieved by means of adaptive changes of the primal and dual stepsizes τ and ρ, as well as of the relaxation parameter θ, which are updated according to the scheme:

$$\theta^k = 1/\sqrt{1+2\gamma\tau^k},\ \tau^{k+1} = \theta^k\tau^k,\ \rho^{k+1} = \rho^k/\theta^k \ ,$$

where $\gamma \leq \sigma_f$, assuming knowledge of the convexity modulus of f. The variant results in a *global $O(1/k)$ convergence rate for the primal sequence* $\{z^k\}$, [19, Theorem 2]. In case that Q is diagonal, the extra computational cost that comes from the acceleration is insignificant.

	Stepsize restrictions	Strong convexity assumptions	Decouples variables of linear constraints	Convergence in function values	Convergence in sequences
ADMM	no	no	no	ergodic $O(1/k)$ [26], [15]	$O(1/\sqrt{k})$ [15], linear [17], [27], [28]
AMA	yes	yes on $f(z)$	yes	-	$O(1/\sqrt{k})$ on the primal [29]
CPI	yes	no	yes	ergodic $O(1/k)$ in partial primal-dual gap [19]	linear [19]
FADMM	no	no	no	$O(1/k^2)$ locally on the dual	-
FAMA	yes	yes on $f(z)$	yes	$O(1/k^2)$ on the dual [23], [29]	$O(1/k)$ on the primal [29]
CPII	yes	yes on $f(z)$	yes	-	$O(1/k)$ on the primal [19]

TABLE I

In Table I we provide an up-to-date report of the existing convergence rates of the methods and their accelerated variants. Wherever a dash '-' appears, it means that there does not exist (or we are not aware of) such a result. In some cases, there might be recent advancements that outperform the results presented here.

IV. EXAMPLES

We demonstrate some of the methods presented in the previous sections with an optimal control problem that involves MPC for tracking of a reference signal. We focus on explaining how to rewrite our problems so that we maximally exploit the ideas presented in Section III.

In this example the linearized model of a Boeing 747-200 (B747) is considered [30]. The model has $n = 12$ states and $m = 17$ inputs and the aim is tracking of a reference signal $r(k)$ for three of the states. We discretize with sampling period $T_s = 0.2s$ and consider in total a signal of 115 setpoints. Firstly, a *steady state target calculator* computes a pair of setpoints $(\delta x_s(k), \delta u_s(k))$ for the aircraft, according to a desired reference signal. Subsequently, an MPC controller is tracking the delivered setpoint. The steady-states are generated by solving a strongly convex dense QP with $n+m = 29$ variables and bound constraints on the inputs [30, Section II,B]. The affine term in the objective is a function of $r(k)$, hence the optimization has to be performed as many times as is the length of the reference signal. The MPC problem is a simple quadratic one, with $Q \succeq 0$ and the same bound constraints on the inputs. The affine term is also time-varying since it is a function of the generated setpoints.

a) Steady state calculator: The problem to solve is

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\theta_s^T H_s \theta_s - h_s(k)^T \theta_s \\ \text{subject to} & \theta_{min} \leq \theta_s \leq \theta_{max} \end{array} , \tag{8}$$

with variables $\theta_s \in \mathbb{R}^{n+m}$ and $H_s \succ 0$. Since the objective is strongly convex, we can use accelerated versions of the methods. To this end, FAMA and CPII are valid options, however, the dense structure of H_s would require a forward backward substitution at each iteration, something that can be avoided. We thus take the Cholesky factorization of H_s, *i.e.*, $H_s = LL^T$, L is lower triangular and invertible and perform a change of basis, $\tilde{\theta}_s = L^T \theta_s$. Now the problem can be reformulated as

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\tilde{\theta}_s^T \tilde{\theta}_s - \tilde{h}_s(k)^T \tilde{\theta}_s \\ \text{subject to} & C\tilde{\theta}_s \leq d \end{array} , \tag{9}$$

with variables $\tilde{\theta}_s \in \mathbb{R}^{n+m}$, $\tilde{h}_s(k) = L^{-1}h_s(k)$. The matrix-vector pair (C,d) describes the polytopic constraints that are now imposed in the place of the simple bound constraints that we had in (8). This is the price paid for eliminating the dense Hessian in the objective. By introducing a slack variable $y = C\tilde{\theta}_s - d$, $y \leq 0$, we can apply FAMA to the modified problem with $f(\tilde{\theta}_s) = \frac{1}{2}\tilde{\theta}_s^T\tilde{\theta}_s - \tilde{h}_s(k)^T\tilde{\theta}_s$, $l(y) = \delta_-(y)$, $T = C$, $t = -d$. For the stepsize we choose $\rho = 1/\lambda_{max}(C^TC)$.

As a second option, we use ADMM with the parameters tuned as in [28] in the same setting. This version achieves linear convergence rate by means of the optimal stepsize selection $\rho = 1/\sqrt{\lambda_{min}(CC^T)\lambda_{max}(CC^T)}$. In our case C is singular and so we consider the smallest nonzero eigenvalue.

Accordingly we can use CPII. Problem 9 can be written in a saddle point form as

$$\min_{\tilde{\theta}_s} \max_{\lambda} \left\{ \left\langle C\tilde{\theta}_s - d, \lambda \right\rangle + \frac{1}{2}\tilde{\theta}_s^T\tilde{\theta}_s - \tilde{h}_s(k)^T\tilde{\theta}_s - \delta_+(\lambda) \right\} ,$$

so we can use CPII with $Z = \mathbb{R}^{n+m}$, $l_i^\star(\lambda) = \delta_+(\lambda)$, T,t as defined above. Note that there are no equality constraints, hence there is no ν-update. We initialize the primal stepsize $\tau^0 = 100$ according to [19, Theorem 2].

We solve the problem 115 times with the affine term varying slightly from one iteration to the other. We terminate based on the residual decrease, with the accuracy threshold set to 10^{-3} for FAMA and CPII and 10^{-4} for ADMM (see Remark 1). FAMA needs 495 iterations on average, with average time 0.85ms per solve, ADMM 194 iterations at 0.56ms per solve and CPII 1100 iterations at 4.9ms per solve. The solutions achieved are quite accurate, with a normed relative error ($\|\theta_s - \theta_s^\star\| / \|\theta_s^\star\|$) of $\approx 10^{-5}$ for all the methods, sumed over all 115 instances. The optimal stepsize selection renders ADMM clearly superior in this case.

b) MPC for tracking: The MPC problem described in [30, Section II] can be written in the condensed form

$$\begin{array}{ll} \text{minimize} & \delta_{u_s}^T \delta_{u_s}^T + h(k)^T \delta_{u_s} \\ \text{subject to} & C\delta_{u_s} \leq d \end{array} , \tag{10}$$

with variables $\delta_{u_s} \in \mathbb{R}^{Nm}$, after having changed the basis in the same way as before. We solve the problem for the following scenarios: $N = 5$, cold start, warm started at the primal and dual optima of the previous solve. The outputs are reported in Table II. ADMM behaves significantly better

		ADMM	FAMA	CPII
$N=5$	Av. No. Iters. Cold\Warm	1362 \548	2279 \778	1544\825
	Min.\Max. No. Iters. Warm	72\1504	83\5947	1\2111
	Av. Time Cold \Warm (ms)	46.90\19.82	42.74 \14.82	75.16 \40.53
	Relative error $\|(x,u)-(x^\star,u^\star)\|/\|((x^\star,u^\star))\|$	1.61×10^{-4}	1.62×10^{-4}	1.61×10^{-4}

TABLE II

than the other two methods in terms of iterations, but FAMA is faster overall in timings. With the number of variables increasing, the cost per iteration starts being more evident when using ADMM. We observe that warm starting makes a big difference in terms of iteration counts.

Remark 1: Termination criteria for all methods have been derived in the spirit of [23, Section 1]. We define primal and dual residuals for ADMM (FADMM) as

$$r^k = -t - Tz + y, \quad s^k = -\rho T^T(y^k - y^{k-1}) \ ,$$

for AMA (FAMA)

$$r^k = -t - Tz + y, \quad s^k = -\rho T^T(\lambda^k - \lambda^{k-1}) \ ,$$

while for CPI and CPII we have accordingly

$$s^k = -\begin{bmatrix} T \\ A \end{bmatrix}(z^k - z^{k+1}) + \frac{1}{\rho}\begin{bmatrix} \lambda^k - \lambda^{k+1} \\ \nu^k - \nu^{k+1} \end{bmatrix}$$

$$r^k = \frac{1}{\tau}(z^k - z^{k+1}) - \begin{bmatrix} T \\ A \end{bmatrix}^T \begin{bmatrix} \lambda^k - \lambda^{k+1} \\ \nu^k - \nu^{k+1} \end{bmatrix} \ .$$

Termination holds whenever $\|r^k\|_2 \leq \varepsilon$ and $\|s^k\|_2 \leq \varepsilon$.

V. CONCLUSIONS

We demonstrated how three popular splitting methods can be derived from a general scheme and we discussed accelerated variants, mostly based on Nesterov's optimal relaxation sequence. The methods were applied to an MPC problem for the control of a Boeing 747 aircraft. A more complete survey including some other methods, techniques for computational speedup as well as more examples will follow soon.

REFERENCES

[1] E. Ullmann, "A Matlab toolbox for C-code generation for first order methods." Master's thesis, ETH Zurich, 2011.

[2] A. Domahidi, A. U. Zgraggen, M. N. Zeilinger, M. Morari, and C. N. Jones, "Efficient interior point methods for multistage problems arising in receding horizon control.," in *CDC*, 2012.

[3] J. Mattingley and S. Boyd, "CVXGEN: a code generator for embedded convex optimization," *Optimization and Engineering*, 2012.

[4] H. J. Ferreau, H. G. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *International Journal of Robust and Nonlinear Control*, 2008.

[5] R. Hestenes, "Multiplier and gradient methods," *Journal of Optimization Theory and Applications*, 1969.

[6] L. H. K. J. Arrow and H. Uzawa, "Studies in linear and non-linear programming," *Stanford University Press*, 1958.

[7] J. Douglas and H. H. Rachford, "On the numerical solution of heat conduction problems in two and three space variables," *Comp. Math. Appl.*, 1956.

[8] R. Glowinski and A. Marrocco, "A Modification of the Arrow-Hurwicz Method for Search of Saddle Points," 1975.

[9] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite-element approximations," *Comp. Math. Appl.*, 1976.

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, 2011.

[11] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*, pp. 185–212, Springer New York, 2011.

[12] J. Esser, *Primal Dual Algorithms for Convex Models and Applications to Image Restoration, Registration and Nonlocal Inpainting*. 2010.

[13] H. H. Bauschke and P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*. Springer Science+ Business Media, 2011.

[14] R. Rockafellar, "Augmented lagrangians and applications of the proximal point algorithm in convex programming," *Mathematics of operations research*, 1956.

[15] R. Shefi and M. Teboulle, "Rate of Convergence Analysis of Decomposition Methods Based on the Proximal Method of Multipliers for Convex Minimization," *SIAM Journal on Optimization*, 2014.

[16] M. B. X. Zhang and S. Osher, "A Unified Primal-Dual Algorithm Framework Based on Bregman Iteration," *Journal of Scientific Computing*, 2011.

[17] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *Rice CAAM technical report TR12-14*, 2012.

[18] P. Tseng, "Applications of splitting algorithm to decomposition in convex programming and variational inequalities," *SIAM J. Control Optim.*, 1991.

[19] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, 2011.

[20] B. O'Donoghue, G. Stathopoulos, and S. Boyd, "A splitting method for optimal control," *IEEE Transactions on Control Systems Technology*, 2012.

[21] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[22] Y. Nesterov, "A method for solving the convex programming problem with convergence rate $O(1/k^2)$," *Dokl. Akad. Nouk SSSR*, 1983.

[23] T. Goldstein, B. O'Donoghue, and S. Setzer, "Fast Alternating Direction Optimization Methods," *arXiv.org*, 2012.

[24] B. O'Donoghue and E. Candes, "Adaptive Restart for Accelerated Gradient Schemes," *arXiv.org*, 2012.

[25] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, 2009.

[26] B. He and X. Yuan, "On the $O(1/n)$ Convergence Rate of the Douglas-Rachford Alternating Direction Method," *SIAM J. Numerical Analysis*, 2012.

[27] Z.-Q. L. Mingyi Hong, "On the linear convergence of the alternating direction method of multipliers," 2012.

[28] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems," *arXiv preprint arXiv:1306.2454*, 2013.

[29] A. Beck and M. Teboulle, "A fast dual proximal gradient algorithm for convex minimization and applications," *Operations Research Letters*, 2014.

[30] E. Hartley, J. Jerez, A. Suardi, J. M. Maciejowski, E. Kerrigan, and G. Constantinides, "Predictive Control using an FPGA with Application to Aircraft Control," *IEEE Transactions on Control Systems Technology*, 2013.