A Parametric Non-Convex Decomposition Algorithm for Real-Time and Distributed NMPC

Jean-Hubert Hours  Colin N. Jones

Abstract

A novel decomposition scheme to solve parametric non-convex programs as they arise in Nonlinear Model Predictive Control (NMPC) is presented. It consists of a fixed number of proximal linearised alternating minimisations and a dual update per time step. Hence, the proposed approach is attractive in a real-time distributed context. Assuming that the Nonlinear Program (NLP) is semi-algebraic and that its critical points are strongly regular, contraction of the sequence of primal-dual iterates is proven, implying stability of the sub-optimality error, under some mild assumptions. Moreover, it is shown that the performance of the optimality-tracking scheme can be enhanced via a continuation technique. The efficacy of the proposed decomposition method is demonstrated by solving a centralised NMPC problem to control a DC motor and a distributed NMPC program for collaborative tracking of unicycles, both within a real-time framework. Furthermore, an analysis of the sub-optimality error as a function of the sampling period is proposed given a fixed computational power.

Index Terms

Augmented Lagrangian, Alternating minimisation, Proximal gradient, Strong regularity, Kurdyka-Lojasiewicz inequality, Nonlinear Model Predictive Control.

Jean-Hubert Hours and Colin N. Jones are with the Laboratoire d’Automatique, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne (e-mail: jean-hubert.hours@epfl.ch, colin.jones@epfl.ch).

The work of Jean-Hubert Hours is supported by the Swiss National Science Foundation (SNSF) under grant 200021-135218. The work of Colin N. Jones is supported by the European Research Council under the European Union’s Seventh Framework Programme (FP/2007-2013), ERC Grant Agreement 307608.
I. INTRODUCTION

The applicability of NMPC to fast and complex dynamics is hampered by the fact that an NLP, which is generally non-convex, is to be solved at every sampling time. Solving an NLP to full accuracy is not tractable when the system’s sampling frequency is high, which is the case for many mechanical or electrical systems. This difficulty is enhanced when dealing with distributed systems, which consist of several sub-systems coupled through their dynamics, objectives or constraints. This class of systems typically lead to large-scale NLPs, which are to be solved online as the system evolves. Solving such programs in a centralised manner may be computationally too demanding and may also hamper autonomy of the agents. Therefore, much research effort is currently brought to develop decentralised computational methods applicable to MPC. Although several distributed linear MPC algorithms are now available, there only exists very few strategies [23], [16] that can address online distributed NMPC programs, as they generally result in non-convex problems. Most of these techniques essentially consist in fully convexifying the problem at hand and resort to Sequential Convex Programming (SCP). Hence, from a theoretical point of view, they cannot be fully considered as distributed non-convex techniques, since the decomposition step occurs at the convex level. In addition, the SCP methods of [23], [16] have not been analysed in an online optimisation framework, where fixing the number of iterations is critical. In general, it should be expected that SCP techniques require a high number communications between agents, since a sequence of distributed convex NLPs is to be solved. Therefore, our main objective is to propose a novel online distributed optimisation strategy for solving NMPC problems in a real-time framework. Our approach is useful for deployment of NMPC controllers, but can also be applied in the very broad parametric optimisation context.

A standard way of solving distributed NLPs is to apply Lagrangian decomposition [9]. This technique requires strong duality, which is guaranteed by Slater’s condition in the convex case, but rarely holds in a non-convex setting. The augmented Lagrangian framework [5] mitigates this issue by closing the duality gap for properly chosen penalty parameters [27]. The combination of the bilinear Lagrangian term with the quadratic penalty also turns out to be computationally more efficient than standard penalty approaches, as the risk of running into ill-conditioning is reduced by the fast convergence of the dual sequence. In addition, global convergence of the dual iterates can be obtained, as in the LANCELOT package [12], [10]. However, the quadratic
penalty term induces non-separability in the objective, which hampers decomposing the NLP completely. Several approaches have been proposed to remedy this issue [20]. Taking inspiration from the Alternating Direction Method of Multipliers (ADMM) [9], which has recently gained in popularity, even in a non-convex setting (yet without convergence guarantees) [21], we propose addressing the non-separability issue via a novel Block-Coordinate Descent (BCD) type technique. BCD or alternating minimisation strategies are known to lead to ‘easily’ solvable subproblems, which can be parallelised under some assumptions on the coupling, and are well-suited to distributed computing platforms [6]. BCD-type techniques are currently raising interest for solving very large-scale programs [24]. Until very recently, for non-convex objectives with certain structure, convergence of the BCD iterates was proven in terms of limit points only [30]. The central idea of our algorithm is to apply a truncated proximal alternating linearised minimisation in order to solve the primal augmented Lagrangian problem approximately. Our convergence analysis is based on the recent results of [2], [8], which provide a very general framework for proving global convergence of descent methods on non-smooth semi-algebraic objectives.

Augmented Lagrangian techniques have proven effective at solving large-scale NLPs [33]. Yet, in an online context, they are hampered by the fact that a sequence of non-convex programs need to be solved to an increasing level of accuracy [5]. Recently, a parametric augmented Lagrangian algorithm has been introduced by [32] for centralised NMPC problems. Our online decomposition method builds upon the ideas of [32], but extends them to the online distributed framework, which requires a significantly different approach. Furthermore, our analysis brings the results of [32] one step further by proving contraction of the sequence of primal-dual iterates. In particular, new insights are given on how the penalty parameter and the number of primal iterations need to be tuned in order to ensure boundedness of the tracking error, which is the core of the analysis of a parametric optimisation algorithm. Moreover, an interesting aspect of our analysis is that it shows how the proposed decomposition algorithm can be efficiently modified via a continuation, or homotopy, strategy [1], leading to faster convergence of the tracking scheme. This theoretical observation is confirmed by a numerical example presented later in the paper. Our technique is also designed to handle a more general class of parametric NLPs, where the primal Projected Successive Over-Relaxation (PSOR) of [32] is limited to quadratic objectives subject to non-negativity constraints.

From an MPC perspective, the effect of the sampling period on the behaviour of the combined
system-optimiser dynamics has not been analysed in a very rigorous manner yet. Therefore, it is widely admitted that faster sampling rates lead to better closed-loop performance. Yet, this thinking neglects the fact that MPC is an optimisation-based control strategy that requires a significant amount of computations within a sampling interval. Hence, sampling faster entails reducing the number of iterations in the optimiser quite significantly, as the computational power of any computing platform is limited, especially in the case of embedded or distributed applications. In the last part of the paper, we propose an analysis of the effect of the sampling period on the behaviour of the system in closed-loop with our parametric decomposition strategy. It is demonstrated that the proposed theory accounts for the observed numerical behaviour quite nicely. In particular, tuning the penalty parameter of the augmented Lagrangian turns out to significantly improve the tracking performance at a given sampling period.

In Section III, the parametric distributed optimisation scheme is presented. In Section IV, some key theoretical ingredients such as strong regularity of generalised equations and the Kurdyka-Lojasiewicz inequality are introduced, and convergence of the primal sequence is proven. Then, in Section V, contraction of the primal-dual sequence is proven and conditions ensuring stability of the tracking error are derived. In Section VI, basic computational aspects are investigated. Finally, the proposed approach is tested on two numerical examples, which consist in controlling the speed of a DC motor to track a piecewise constant reference for the first one, and collaborative tracking of unicycles for the second one. In the case of the DC motor, the effect of the sampling period on the tracking error, passed through the system dynamics, is analysed given a fixed computational power.

II. BACKGROUND

**Definition 1** (Proximal operator). Let \( f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\} \) be a proper lower-semicontinuous function and \( \alpha > 0 \). The proximal operator of \( f \) with coefficient \( \alpha \), denoted by \( \text{prox}_\alpha (\cdot) \), is defined as follows:

\[
\text{prox}_\alpha (x) := \arg\min_y f(y) + \frac{\alpha}{2} \| y - x \|_2^2 .
\]  

**Definition 2** (Critical point). Let \( f \) be a proper lower semicontinuous function. A necessary condition for \( x^* \) to be a minimiser of \( f \) is that

\[
0 \in \partial f (x^*) ,
\]
where \( \partial f(x^*) \) is the sub-differential of \( f \) at \( x^* \) [28].

Points satisfying (2) are called critical points.

**Definition 3** (Normal cone to a convex set). Let \( \Omega \) be a convex set in \( \mathbb{R}^n \) and \( \bar{x} \in \Omega \). The normal cone to \( \Omega \) at \( \bar{x} \) is the set
\[
N_\Omega(\bar{x}) := \{ v \in \mathbb{R}^n \mid \forall x \in \Omega, v^\top (x - \bar{x}) \leq 0 \}.
\]

(3)

The indicator function of a closed subset \( \Omega \) of \( \mathbb{R}^n \) is denoted by \( \iota_\Omega \) and is defined as
\[
\iota_\Omega(x) = \begin{cases} 
0 & \text{if } x \in \Omega \\
\infty & \text{if } x \notin \Omega.
\end{cases}
\]

(4)

**Lemma 1** (Sub-differential of indicator function [28]). Given a convex set \( \Omega \), for all \( x \in \Omega \),
\[
\partial \iota_\Omega(x) = N_\Omega(x).
\]

(5)

**Lemma 2** (Descent lemma,[5]). Let \( L : \mathbb{R}^n \to \mathbb{R} \) a continuously differentiable function such that its gradient \( \nabla L \) is \( \lambda_L \)-Lipschitz continuous. For all \( x, y \in \mathbb{R}^n \),
\[
L(y) \leq L(x) + \nabla L(x)^\top (y - x) + \frac{\lambda_L}{2} \|y - x\|^2.
\]

(6)

The distance of a point \( x \in \mathbb{R}^n \) to a subset \( \Sigma \) of \( \mathbb{R}^n \) is defined by
\[
d(x, \Sigma) := \inf_{y \in \Sigma} \|x - y\|_2.
\]

(7)

The open ball with center \( x \) and radius \( r \) is denoted by \( B(x, r) \). Given \( \underline{x}, \bar{x} \in \mathbb{R}^n \), the box set \( \{ x \in \mathbb{R}^n \mid \underline{x} \leq x \leq \bar{x} \} \) is denoted by \( B(\underline{x}, \bar{x}) \). Given a closed convex set \( \Omega \subseteq \mathbb{R}^n \), the single-valued projection onto \( \Omega \) is denoted by \( \pi_\Omega(\cdot) \).
III. SOLVING TIME-DEPENDENT DISTRIBUTED NONLINEAR PROGRAMS

A. Problem formulation

The following class of parametric NLPs is considered:

\[
\begin{align*}
\text{minimise} & \quad J(z, s_k) := Q_J(z, s_k) + \sum_{i=1}^{P} J_i(z_i, s_k) \\
\text{s.t.} & \quad Q_C(z, s_k) = 0, \\
& \quad g_i(z_i, s_k) = 0, \\
& \quad z_i \in Z_i, \ i \in \{1, \ldots, P\},
\end{align*}
\]

where \( z := (z_1^\top, \ldots, z_P^\top)^\top \in \mathbb{R}^{n_z} \), with \( n_z = \sum_{i=1}^{P} n_i \) and \( z_i \in \mathbb{R}^{n_i} \). The vectors \( z_i \) model different agents, while the functions \( Q_J(\cdot, s_k) \) and \( Q_C(\cdot, s_k) : \mathbb{R}^{n_z} \to \mathbb{R}^m \) represent cost and constraint couplings respectively. The functions \( J_i(\cdot, s_k) \) and \( g_i(\cdot, s_k) : \mathbb{R}^{n_z} \to \mathbb{R}^{q_i} \) are individual cost and constraint functionals for agent \( i \). We define \( G(z, s) := (Q_C(z, s)^\top, g_1(z_1, s)^\top, \ldots, g_P(z_P, s)^\top)^\top \).

The constraint sets \( Z_i \) are assumed to be compact and convex, and \( s_k \) is a time-dependent parameter, which lies within a convex set \( S \subseteq \mathbb{R}^p \), where \( k \) is a time index. Primal-dual critical points of NLP (8) are denoted by \( w^*_k \) or \( w^*(s_k) \) without distinction.

**Assumption 1.** The functions \( Q_J, Q_C, J_i \) and \( g_i \) are twice continuously differentiable and semi-algebraic.

B. A non-convex decomposition scheme for optimality tracking

At every time instant \( k \), a critical point of the parametric NLP (8) is computed inexactly in a distributed manner. The key idea is to track time-dependent optima \( w^*_k \) of (8) by approximately computing saddle points of the augmented Lagrangian

\[
L_\rho(z, \mu, s_k) := J(z, s_k) + \left( \mu + \frac{\rho}{2} G(z, s_k) \right)^\top G(z, s_k),
\]

subject to \( z \in Z \), where \( Z := Z_1 \times Z_2 \times \ldots \times Z_P \) and \( \mu := (\mu_C^\top, \mu_1^\top, \ldots, \mu_P^\top)^\top \in \mathbb{R}^{m+q} \), with \( q := \sum_{i=1}^{P} q_i \), is a dual variable associated with the equality constraints \( Q_C(z, s_k) = 0, g_1(z_1, s_k) = 0, \ldots, g_P(z_P, s_k) = 0 \). The scalar \( \rho > 0 \) is the so-called **penalty parameter**. In the rest of the paper, sub-optimality of a variable is highlighted by a \( \bar{\cdot} \), and criticality by a \( \cdot^* \). We
Algorithm 1 Optimality tracking splitting algorithm

**Input:** Suboptimal primal-dual solution $(\bar{z}_k^T, \bar{\mu}_k^T)^T$, state-parameter $s_{k+1}$, augmented Lagrangian $L_\rho (\cdot, \bar{\mu}_k, s_{k+1})$.

**Primal/Inner loop:**

$z^{(0)} \leftarrow \bar{z}_k$

$c_i^{(-1)} \leftarrow c_{i_{\text{smallest}}}^{\text{sm}}$ for $i \in \{1, \ldots, P\}$

for $l = 0 \ldots M - 1$ do

for $i = 1 \ldots P$ do

$(z_i^{(l+1)}, c_i^{(l)}) \leftarrow \text{bckMin} \left( z_i^{(l)}, c_i^{(l-1)}, \bar{\mu}_k, \rho, s_{k+1} \right)$

end for

end for

$\bar{z}_{k+1} \leftarrow z^{(M)}$

**Dual update:** $\bar{\mu}_{k+1} \leftarrow \bar{\mu}_k + \rho G (\bar{z}_{k+1}, s_{k+1})$

intend to build an approximate solution $(\bar{z}_{k+1}^T, \bar{\mu}_{k+1}^T)^T$ to (8) by incremental improvement from $(\bar{z}_k^T, \bar{\mu}_k^T)^T$, once the state estimate $s_{k+1}$ has been received.

**Remark 1.** Incremental approaches are broadly applied in NMPC, as fully solving an NLP takes a significant amount of time and may result in unacceptable time delays. Yet, existing incremental NMPC strategies [15], [34] are based on Newton predictor-corrector steps, which require factorisation of a KKT system. This a computationally demanding task for large-scale systems that cannot be readily carried out in a distributed context. Therefore, Algorithm 1 can be interpreted as a distributed incremental improvement technique for NMPC.

We define the partial augmented Lagrangian function at block $i$

$$L_{\rho, \mu, s}^{(i)} := L_\rho (z_1, \ldots, z_{i-1}, \cdot, z_{i+1}, \ldots, z_P, \mu, s)$$

and the quadratic model at $z_i$ given a curvature coefficient $c_i$

$$q (z; z_i, c_i) := L_{\rho, \mu, s}^{(i)} (z_i) + \nabla L_{\rho, \mu, s}^{(i)} (z_i)^T (z - z_i) + \frac{c_i}{2} \| z - z_i \|_2^2 .$$

Given an iteration index $l$, we define

$$L_{\rho, \mu, s}^{(i,l)} := L_\rho (z_1^{(l+1)}, \ldots, z_{i-1}^{(l+1)}, \cdot, z_{i+1}^{(l)}, \ldots, z_P^{(l)}, \mu, s) .$$
Algorithm 2 Backtracking procedure at agent $i$ and iteration $l$, $bckMin(z_i, c_i, \mu, \rho, s)$

**Input:** Primal variable $z_i \in \mathbb{Z}_i$, curvature estimate $c_i$, partial augmented Lagrangian $L_{\rho,\mu,s}(\cdot)$ and quadratic model $q(\cdot; z_i^{(c)}, c_i)$.

$z_i^{(c)} \leftarrow z_i$

$z_i^{(u)} \leftarrow z_i^{(c)}$

**Backtracking loop:**

While $L_{\rho,\mu,s}(z_i^{(u)}) + \frac{\alpha_i}{2} \| z_i^{(u)} - z_i^{(c)} \|_2^2 > q(z_i^{(u)}, z_i^{(c)}, c_i)$ do

$c_i \leftarrow \beta \cdot c_i$

$z_i^{(u)} \leftarrow \text{prox}_{\delta Z_i} \left( z_i^{(c)} - \frac{1}{c_i} \nabla L_{\rho,\mu,s}(z_i^{(c)}) \right)$

end while

$z_i \leftarrow z_i^{(u)}$

Algorithm 1 computes a suboptimal primal variable $\bar{z}_{k+1}$ by applying $M$ iterations of a proximal alternating linearised method to minimise the augmented Lagrangian functional $L_{\rho}(\cdot, \bar{\mu}_k, s_{k+1}) + \sum_{i=1}^{P} \delta Z_i(\cdot)$. Each step of the proximal alternating minimisation consists in proximal back-tracking minimisations, presented in Algorithms 1 and 2. Later in the paper, it is proven that Algorithm 2 terminates and guarantees convergence of the whole procedure to a critical point of the augmented Lagrangian (9) for an infinite number of primal iterations ($M = \infty$ in Algorithm 1). In practice, after a fixed number of primal iterations $M$, the dual variable is updated in a first-order fashion. The whole procedure yields a suboptimal primal-dual point $\bar{w}_{k+1} = (\bar{z}_{k+1}^T, \bar{\mu}_{k+1}^T)^T$ for program (8) given parameter $s_{k+1}$.

**Remark 2.** The conservatism of $bckMin$ can be reduced if the sets $Z_i$ are decomposable into ‘nice’ convex sets, meaning that larger step-sizes $\frac{1}{c_i}$ can be obtained.

**Remark 3.** Note that the active-set at $z_{k+1}^*$ may be different from the active-set at $z_k^*$. Hence, Algorithm 1 should be able to detect active-set changes quickly. This is the role of the proximal steps, where projections onto the sets $Z_i$ are carried out. It is well-known that gradient projection methods allow for fast activity detection [11].
IV. THEORETICAL TOOLS: STRONG REGULARITY AND KURDYKA-LOJASIEWICZ INEQUALITY

The analysis of the splitting Algorithm 1 is based on the concept of generalised equations, which has been introduced in real-time optimisation by [32]. Another key ingredient for the convergence of the proximal alternating minimisations in Algorithm 1 is the Kurdyka-Lojasiewicz property, which has been introduced in nonlinear programming by [2], [8] and is satisfied by semi-algebraic and real sub-analytic functions [7], which encompass a broad class of functions appearing in NLPs arising from the discretisation of optimal control problems via, for instance, collocation methods [19].

A. Parametric generalised equations

Critical points \( w^*(s) = (z^*(s)^\top, \mu^*(s)^\top)^\top \) of the parametric nonlinear program (8) satisfy \( z^*(s) \in \mathcal{Z} \) and

\[
\begin{align*}
0 & \in \nabla_z J(z^*(s), s) + \nabla_z G(z^*(s), s)^\top \mu^*(s) + \mathcal{N}_{\mathcal{Z}}(z^*(s)) \\
G(z^*(s), s) & = 0
\end{align*}
\]  

Relation (13) can be re-written as the generalised equation

\[
0 \in F(w, s) + \mathcal{N}_{\mathcal{Z} \times \mathbb{R}^m}(w),
\]

where

\[
F(w, s) := \begin{bmatrix} \nabla_z J(z, s) + \nabla_z G(z, s)^\top \mu \\ G(z, s) \end{bmatrix}, \quad w = \begin{bmatrix} z \\ \mu \end{bmatrix}.
\]

In order to analyse the behaviour of the critical points of (8) as the parameter \( s_k \) evolves over time, the generalised equation (14) should satisfy some regularity assumptions. This is captured by the strong regularity concept [26], which has been first applied in an NMPC context by [32].

**Definition 4** (Strong regularity,[26]). Let \( C \) be a compact convex set in \( \mathbb{R}^n \) and \( f: \mathbb{R}^n \to \mathbb{R}^n \) a differentiable mapping. A generalised equation \( 0 \in f(x) + \mathcal{N}_C(x) \) is said to be strongly regular at a solution \( x^* \in C \) if there exists radii \( \eta > 0 \) and \( \kappa > 0 \) such that for all \( r \in \mathcal{B}(0, \eta) \), there exits a unique \( x_r \in \mathcal{B}(x^*, \kappa) \) such that

\[
r \in f(x^*) + \nabla f(x^*)(x_r - x^*) + \mathcal{N}_C(x_r),
\]
and the inverse mapping \( r \mapsto x_r \) from \( \mathcal{B}(0, \eta) \) to \( \mathcal{B}(x^*, \kappa) \) is Lipschitz continuous.

**Remark 4.** Note that strong regularity incorporates active-set changes in its definition, as the normal cone is taken at \( x_r \) in Eq. (16). The set of active constraints at \( x_r \) may be different from the one at \( x^* \), but Lipschitz continuity of the solution is preserved.

**Remark 5.** When the constraint set \( Z \) is polyhedral, it can be shown that strong regularity is equivalent to linear independence constraints qualification and strong second-order optimality [17].

As parameter \( s_k \) changes in time, strong regularity is assumed at every time instant \( k \).

**Assumption 2.** For all parameters \( s_k \in \mathcal{S} \) and associated solutions \( w_k^* \), the generalised equation (14) is strongly regular at \( w_k^* \).

From the strong regularity Assumption 2, it can be proven that the non-smooth manifold formed by the solutions to the parametric program (8) is locally Lipschitz continuous. The first step to achieve this fundamental property is the following Theorem proven in [26].

**Theorem 1.** There exists radii \( \delta_A > 0 \) and \( r_A > 0 \) such that for all \( k \in \mathbb{N} \), for all \( s \in \mathcal{B}(s_k, r_A) \), there exists a unique \( w^*(s) \in \mathcal{B}(w_k^*, \delta_A) \) such that

\[
0 \in F(w^*(s), s) + N_{\mathcal{Z} \times \mathbb{R}^m}(w^*(s))
\]

and for all \( s, s' \in \mathcal{B}(s_k, r_A) \),

\[
\|w^*(s) - w^*(s')\|_2 \leq \lambda_A \|F(w^*(s'), s) - F(w^*(s'), s')\|_2,
\]

where \( \lambda_A \) is a Lipschitz constant associated with the strong regularity mapping of (14).

**Remark 6.** Theorem 1 is actually a refinement of Theorem 2.1 in [26], as the radii \( \delta_A \) and \( r_A \) are assumed not to depend on the parameter \( s_k \in \mathcal{S} \).

Relation (18) does not exactly correspond to a Lipschitz property, but this is easily fixed by strengthening the continuity of the mapping \( F \) in the parameter \( s \).

**Assumption 3.** There exists \( \lambda_F > 0 \) such that for all \( w \in \mathcal{Z} \times \mathbb{R}^m \),

\[
\forall s, s' \in \mathcal{S}, \|F(w, s) - F(w, s')\|_2 \leq \lambda_F \|s - s'\|_2.
\]
In an NMPC context, Assumption 3 is always satisfied with $\lambda_F = 1$, as a constraint of the form $x_0 - s = 0$, where $x_0$ is the initial state of the prediction horizon, can be enforced in the parametric NLP resulting from the discretisation of the optimal control problem, and a reference in the cost function can be incorporated in the constraints in the same way.

Algorithm 1 tracks the non-smooth solution manifold by traveling from neighbourhood to neighbourhood, where Lipschitz continuity of the primal-dual solution holds. Such tracking procedures have been analysed thoroughly in the unconstrained case by [15] for a Newton-type method, in the constrained case by [32] for an augmented Lagrangian approach and in [29] for an adjoint-based technique. These previous tracking strategies are purely centralised second-order strategies and do not readily extend to solving NLPs in a distributed manner. Our Algorithm 1 proposes a novel way of computing predictor steps along the solution manifold via a decomposition approach, which is tailored to convex constraint sets with closed-form proximal operators. Such a class encompasses boxes, non-negative orthant, semi-definite cones and balls for instance. The augmented Lagrangian framework is particularly attractive in this context, as it allows one to preserve ‘nice’ constraints via partial penalisation.

B. Convergence of the inner loop

The primal loop of Algorithm 1 consists of alternating proximal gradient steps. In general, for non-convex programs as they appear in NMPC, the convergence of such Gauss-Seidel type methods to critical points of the objective is not guaranteed even for smooth functions, as cyclic behaviours may occur [25]. Yet, some powerful convergence results on alternating minimisation techniques have been recently derived [3], [8]. The two key ingredients are coordinate-wise proximal regularisations [2] and the Kurdyka-Lojasiewicz (KL) property [7]. The former enforces a sufficient decrease in the objective, which intuitively prevents the generated sequence from cycling, while the latter ensures a finite-length of the sequence by coupling sufficient decrease with some properties of the sub-differential of the proximal point sequence. In our real-time analysis, the convergence rate of the primal sequence generated by Algorithm 1 is of primary importance.

**Proposition 1** (KL inequality). A proper lower semi-continuous function $L : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ satisfies the KL property at a point $x^*$ in its domain if there exists a neighbourhood $U$ of $x^*$,
\[ \eta \in (0, +\infty) \text{ and } \phi : [0, \eta) \to \mathbb{R}_+ \text{ such that } \phi(0) = 0, \phi \text{ is } C^1 \text{ on } (0, \eta) \text{ with } \phi' > 0 \text{ and } \]

\[ \phi'(L(x) - L(x^*)) d(0, \partial L(x)) \geq 1 , \tag{20} \]

for all \( x \in U \cap \{ x \in \mathbb{R}^n \mid L(x^*) < L(x) < L(x^*) + \eta \} \).

A semi-algebraic function \( L \) satisfies the KL inequality at any of its critical points \( x^* \) with \( \phi(t) = ct^{1-\theta} \), as proven in [7]. More precisely, there exists \( \delta > 0, c > 0 \) and \( \theta \in [0, 1) \) such that for all \( x \in B(x^*, \delta) \cap \{ x \in \mathbb{R}^n \mid L(x) > L(x^*) \} \),

\[ d(0, \partial L(x)) \geq c (L(x) - L(x^*))^{\theta} . \tag{21} \]

The parameter \( \theta \), also called the Lojasiewicz exponent, can be interpreted as a shape parameter of the graph of \( L \) around its critical point \( x^* \). When \( \theta \) is close to 0, the graph is sharp around \( x^* \), whereas when \( \theta \) is close to 1, the graph is flat around \( x^* \).

**Assumption 4.** For all \( \mu \in \mathbb{R}^{m+q} \) and \( s \in S \), the augmented Lagrangian (9) satisfies the KL property (21) with Lojasiewicz exponents \( \theta(\mu, s) \in (1/2, 1) \) and radius \( \delta > 0 \) at all its critical points. The exponents \( \theta(\mu, s) \) are upper bounded by \( \hat{\theta} \in (1/2, 1) \).

It can be proven that for real analytic functions, the Lojasiewicz exponent lies within \([1/2, 1)\) [22]. Moreover, for multivariate polynomials of degree higher than two, such as \( L_\rho(\cdot, \mu, s) \), an upper bound on the Lojasiewicz exponent can be derived, which only depends on the variable dimension and the degree [13]. For strongly convex functions \( \delta = +\infty \), which shows that the KL inequality 1 does not restrict the convergence analysis to a very small neighbourhood of the critical point.

In order two ensure convergence of the primal loop of Algorithm 1 via Theorem 2.9 in [4], two ingredients are needed: a sufficient decrease property and a relative error condition. From the sufficient decrease, convergence of the series \( \sum \| z^{(l+1)} - z^{(l)} \|_2^2 \) is readily deduced. By combining the relative error condition and the KL property, this can be turned into convergence of the series \( \sum \| z^{(l+1)} - z^{(l)} \|_2 \), ensuring convergence of the sequence \( \{ z^{(l)} \} \) via a Cauchy sequence argument.

**Lemma 3** (Sufficient decrease in the primal). For all \( l \geq 0, \mu \in \mathbb{R}^{m+q}, s \in S \) and \( \rho > 0 \),

\[ L_\rho(z^{(l+1)}, \mu, s) + t_\mathcal{E}(z^{(l+1)}) + \frac{\alpha}{2} \| z^{(l+1)} - z^{(l)} \|_2^2 \leq L_\rho(z^{(l)}, \mu, s) + t_\mathcal{E}(z^{(l)}) , \tag{22} \]
where \( \alpha := \min \{\alpha_i \mid i \in \{1, \ldots, P\}\} \).

**Proof:** We first need to show that the backtracking procedure described in Algorithm 2 terminates. This is an almost direct consequence of the Lipschitz continuity of the gradient of \( L_{\rho,\mu,s}^{(i)} \) for \( i \in \{1, \ldots, P\} \), as the augmented Lagrangian is twice continuously differentiable and the constraints sets \( Z_i \) are compact. From Lemma 2, it follows that for all \( i \in \{1, \ldots, P\} \) and \( l \geq 0 \),

\[
L_{\rho,\mu,s}^{(i)}\left(z_i^{(l+1)}\right) \leq L_{\rho,\mu,s}^{(i)}\left(z_i^{(l)}\right) + \nabla L_{\rho,\mu,s}^{(i)}\left(z_i^{(l)}\right)^{\top}\left(z_i^{(l+1)} - z_i^{(l)}\right) + \frac{\lambda_i}{2}\left\|z_i^{(l+1)} - z_i^{(l)}\right\|_2^2 ,
\]

(23)

where \( \lambda_i \) is a Lipschitz constant of \( \nabla L_{\rho,\mu,s}^{(i)} \). By taking \( c_i > \lambda_i + \alpha_i \), which is satisfied at some point in the while loop of Algorithm 2, one gets

\[
L_{\rho,\mu,s}^{(i)}\left(z_i^{(l+1)}\right) + \frac{\alpha_i}{2}\left\|z_i^{(l+1)} - z_i^{(l)}\right\|_2^2 \leq L_{\rho,\mu,s}^{(i)}\left(z_i^{(l)}\right) + \nabla L_{\rho,\mu,s}^{(i)}\left(z_i^{(l)}\right)^{\top}\left(z_i^{(l+1)} - z_i^{(l)}\right) + \frac{c_i}{2}\left\|z_i^{(l+1)} - z_i^{(l)}\right\|_2^2 ,
\]

(24)

which is exactly the termination criterion of the while loop in Algorithm 2. Moreover,

\[
z_i^{(l+1)} = \text{prox}_{\delta c_i}\left(z_i^{(l)} - \frac{1}{c_i}\nabla L_{\rho,\mu,s}^{(i)}\left(z_i^{(l)}\right)\right)
= \arg\min_{z \in Z_i} \frac{c_i}{2}\left\|z - \left(z_i^{(l)} - \frac{1}{c_i}\nabla L_{\rho,\mu,s}^{(i)}\left(z_i^{(l)}\right)\right)\right\|_2^2
\]

\[
= \arg\min_{z \in Z_i} \nabla L_{\rho,\mu,s}^{(i)}\left(z_i^{(l)}\right)^{\top}\left(z - z_i^{(l)}\right) + \frac{c_i}{2}\left\|z - z_i^{(l)}\right\|_2^2 .
\]

(25)

Hence

\[
\nabla L_{\rho,\mu,s}^{(i)}\left(z_i^{(l)}\right)^{\top}\left(z_i^{(l+1)} - z_i^{(l)}\right) + \frac{c_i}{2}\left\|z_i^{(l+1)} - z_i^{(l)}\right\|_2^2 \leq 0 ,
\]

(26)

which implies that

\[
L_{\rho,\mu,s}^{(i)}\left(z_i^{(l+1)}\right) + \alpha_i\left\|z_i^{(l+1)} - z_i^{(l)}\right\|_2^2 \leq L_{\rho,\mu,s}^{(i)}\left(z_i^{(l)}\right) + \alpha_i\left\|z_i^{(l+1)} - z_i^{(l)}\right\|_2^2 ,
\]

(27)

as \( z_i^{(l+1)}, z_i^{(l)} \in Z_i \). Combining equalities (27) for all \( i \in \{1, \ldots, P\} \), one obtains the sufficient decrease property (22).

**Lemma 4** (Relative error condition). For all \( \mu \in \mathbb{R}^{m+q} \) and all \( s \in S \), there exists \( \gamma (\mu, s) > 0 \) such that

\[
\exists v^{(l+1)} \in N_Z \left(z^{(l+1)}\right), \left\|\nabla_z L_\mu \left(z^{(l+1)}\right, \mu, s\right) + v^{(l+1)}\right\|_2 \leq \gamma (\mu, s) \left\|z^{(l+1)} - z^{(l)}\right\|_2 .
\]

(28)
Proving: From the definition of $z_i^{(l+1)}$ as a proximal iterate, we have that for all $i \in \{1, \ldots, P\}$,

$$\exists v_i^{(l+1)} \in \mathcal{N}_{z_i} \left( z_i^{(l+1)} \right), \quad 0 = \nabla L_{\rho, \mu, s}^{(i)} \left( z_i^{(l)} \right) + c_i \left( z_i^{(l+1)} - z_i^{(l)} \right) + v_i^{(l+1)},$$

(29)

Hence

$$0 = \nabla L_{\rho, \mu, s}^{(i)} \left( z_i^{(l+1)} \right) + \nabla L_{\rho, \mu, s}^{(i)} \left( z_i^{(l)} \right) - \nabla L_{\rho, \mu, s}^{(i)} \left( z_i^{(l+1)} \right) + c_i \left( z_i^{(l+1)} - z_i^{(l)} \right) + v_i^{(l+1)},$$

(30)

and from the Lipschitz continuity of $\nabla L_{\rho, \mu, s}^{(i)}$, one immediately obtains

$$\left\| v_i^{(l+1)} + \nabla L_{\rho, \mu, s} \left( z_i^{(l+1)} \right) \right\|_2 \leq (\lambda_i + c_i) \left\| z_i^{(l+1)} - z_i^{(l)} \right\|_2.$$ 

(31)

Let $v := (v_1^T, \ldots, v_P^T)^T$. It then follows that

$$\left\| v^{(l+1)} + \nabla L_{\rho, \mu, s} \left( z^{(l+1)} \right) \right\|_2 \leq \sum_{i=1}^{P} \left\| v_i^{(l+1)} + \nabla L_{\rho, \mu, s}^{(i)} \left( z_i^{(l+1)} \right) \right\|_2$$

$$+ \left\| \nabla L_{\rho, \mu, s} \left( z^{(l+1)} \right) - \nabla L_{\rho, \mu, s}^{(i)} \left( z_i^{(l+1)} \right) \right\|_2$$

$$\leq \sum_{i=1}^{P} (\lambda_i + c_i) \left\| z_i^{(l+1)} - z_i^{(l)} \right\|_2 + \lambda \left\| z^{(l+1)} - z^{(l)} \right\|_2$$

$$\leq \left( \sum_{i=1}^{P} (\lambda_i + c_i + \lambda) \right) \left\| z^{(l+1)} - z^{(l)} \right\|_2,$$

which yields the relative error condition (28).

Theorem 2 (Convergence of the primal sequence). Taking $M = \infty$ in Algorithm 1, the primal sequence $\{z^{(l)}\}$ converges to a critical point $z^\infty (\bar{\mu}_k, s_{k+1})$ of $L_{\rho} (\cdot, \bar{\mu}_k, s_{k+1}) + \nu_{\mathcal{Z}} (\cdot)$.

Proof: As $L_{\rho} (\cdot, \bar{\mu}_k, s_{k+1}) + \nu_{\mathcal{Z}} (\cdot)$ satisfies the KL property (Assumption 4), by Lemma 3 and 4, global converge is a direct consequence of Theorem 1 in [8], since $\{z^{(l)}\}$ is bounded.

C. Convergence rate of the primal loop

The results of [2] and [8] provide an asymptotic convergence rate estimate for the proximal alternating loop in Algorithm 1, which depends on the Lojasiewicz exponent, a descriptor of the geometry of the augmented Lagrangian around its critical points.

Lemma 5 (Asymptotic convergence rate estimate). There exists a constant $C > 0$ such that, assuming $\bar{z}_k \in \mathcal{B} \left( 0, \delta \right)$, where $\delta$ has been defined in Proposition 1,

$$\left\| \bar{z}_{k+1} - z^\infty (\bar{\mu}_k, s_{k+1}) \right\|_2 \leq CM^{-\psi(\delta)} \left\| \bar{z}_k - z^\infty (\bar{\mu}_k, s_{k+1}) \right\|_2,$$

(33)
where given \( \theta \in (\frac{1}{2}, 1) \), the function \( \psi \) is defined by
\[
\psi (\theta) := \frac{1 - \theta}{2\theta - 1}.
\]  

**Proof:** As the Lojasiewicz exponent \( \theta (\bar{\mu}_k, s_{k+1}) \) of \( L_\rho (\cdot, \bar{\mu}_k, s_{k+1}) + \iota_Z (\cdot) \) associated with the critical point \( z^\infty (\bar{\mu}_k, s_{k+1}) \) lies within \( (\frac{1}{2}, 1) \) (Assumption 4) and \( \bar{z}_k \in B (0, \delta) \), from [2] and [8], there exists a constant \( C (\bar{\mu}_k, s_{k+1}) > 0 \) such that
\[
\parallel \bar{z}_{k+1} - z^\infty (\bar{\mu}_k, s_{k+1}) \parallel_2 \leq C (\bar{\mu}_k, s_{k+1}) M^{-\psi (\theta (\bar{\mu}_k, s_{k+1}))}.
\]  

One can notice that \( \theta \mapsto M^{-\psi (\theta)} \) is strictly increasing on \( \frac{1}{2} \). Hence, as \( \theta (\bar{\mu}_k, s_{k+1}) \leq \hat{\theta} \) (Assumption 4),
\[
M^{-\psi (\theta (\bar{\mu}_k, s_{k+1}))} \leq M^{-\psi (\hat{\theta})}.
\]  

As \( \bar{z}_k \) is a suboptimal primal solution to the nonlinear program (8) for parameter \( s_k \), it is reasonable to assume that it is not a critical point of \( L_\rho (\cdot, \bar{\mu}_k, s_{k+1}) + \iota_Z (\cdot) \), so that one can claim that there exists a constant \( \kappa > 0 \) such that for all \( k \geq 0 \),
\[
\parallel \bar{z}_k - z^\infty (\bar{\mu}_k, s_{k+1}) \parallel_2 \geq \kappa.
\]  

Then, there exists \( C' (\bar{\mu}_k, s_{k+1}) > 0 \) such that
\[
\parallel \bar{z}_{k+1} - z^\infty (\bar{\mu}_k, s_{k+1}) \parallel_2 \leq C' (\bar{\mu}_k, s_{k+1}) M^{-\psi (\hat{\theta})} \parallel \bar{z}_k - z^\infty (\bar{\mu}_k, s_{k+1}) \parallel_2.
\]  

Without loss of generality, it can be assumed that the constants \( C' (\bar{\mu}_k, s_{k+1}) \) are upper bounded by a constant \( C' \), resulting in (33).

**Remark 7.** The R-convergence rate estimate of Lemma 5 shows that the convergence of the primal sequence \( \{z^{(l)}\} \) is theoretically sub-linear. However, reasonable performance can be observed in practice. Moreover, in this paper, the convergence rate is used only for a theoretical purpose.

V. CONTRACTION OF THE PRIMAL-DUAL SEQUENCE

Algorithm 1 is a truncated scheme both in the primal and dual space, as only \( M \) primal proximal iterations are applied, which are followed by a single dual update. By using warm-starting, it is designed to track the non-smooth solution manifold of the NMPC program. At
a given time instant $k$, the primal-dual solution $\bar{w}_k$ is suboptimal. Thus, a natural question is whether the sub-optimality gap remains stable, as the parameter $s_k$ varies over time, that is if the sub-optimal iterate remains close to the KKT manifold, or converges to it. Intuitively, one can guess that if $s_k$ evolves slowly and the number of primal iterations $M$ is large enough, stability of the sub-optimality error is expected. This section provides a formal statement about the sub-optimality gap and demonstrates that its evolution is governed by the penalty parameter $\rho$, the number of primal iterations $M$ and the magnitude of the parameter difference $s_{k+1} - s_k$, which need to be carefully chosen according to the results provided later in the paper.

A. Existence and uniqueness of critical points

As the overall objective is to analyse the stability of the sub-optimality error $\|\bar{w}_k - w^*_k\|_2$, a unique critical point $w^*_k$ should be defined at every time instant $k$. This is one of the roles of strong regularity. Given a critical point $w^*_k$ for problem (8) at $s_k$, its strong regularity (Assumption 2) implies that there exists a unique critical point for problem (8) at $s_{k+1}$, assuming $\|s_{k+1} - s_k\|_2$ is small enough.

**Assumption 5.** For all $k \geq 0$, $\|s_{k+1} - s_k\|_2 \leq r_A$.

**Lemma 6.** For all $k \geq 0$ and $s_k \in S$, given $w^*_k$ such that

$$0 \in F(w^*_k, s_k) + \mathcal{N}_{Z \times \mathbb{R}^m}(w^*_k),$$

there exists a unique $w^*_{k+1} \in B(w^*_k, \delta_A)$ such that

$$0 \in F(w^*_{k+1}, s_{k+1}) + \mathcal{N}_{Z \times \mathbb{R}^m}(w^*_{k+1}).$$

**Proof:** This is an immediate consequence of Assumption 5 and strong regularity of $w^*_k$ for all $k \geq 0$. 

B. An auxiliary generalised equation

In Algorithm 1, the primal loop, which is initialised at $\bar{z}_k$, converges to $z^\infty(\bar{\mu}_k, s_{k+1})$, a critical point of $L_\rho(\cdot, \bar{\mu}_k, s_{k+1}) + t_Z(\cdot)$, by Theorem 2 in Section IV. The following generalised equation characterises critical points of the augmented Lagrangian function $L_\rho(\cdot, \bar{\mu}, s) + t_Z(\cdot)$
in a primal-dual manner:

$$0 \in H_\rho(w, d_\rho(\bar{\mu}), s) + \mathcal{N}_{Z \times \mathbb{R}^m}(w),$$  \hspace{1cm} (40)

where, given $\mu^*_k$, one defines $d_\rho(\bar{\mu}) := (\bar{\mu} - \mu^*_k) / \rho$ and

$$H_\rho(w, d_\rho(\bar{\mu}), s) := \begin{bmatrix} \nabla z J(z) + \nabla z G(z, s)^\top \mu \\ G(z, s) + d_\rho(\bar{\mu}) + \frac{\mu^*_k - \mu}{\rho} \end{bmatrix}. \hspace{1cm} (41)$$

**Lemma 7.** Let $\bar{\mu} \in \mathbb{R}^m$, $\rho > 0$ and $s \in S$. The primal point $z^*(\bar{\mu}, s)$ is a critical point of $L_\rho(\cdot, \bar{\mu}, s) + \iota_Z(\cdot)$ if and only if the primal-dual point

$$w^*(d_\rho(\bar{\mu}), s) = \begin{bmatrix} z^*(\bar{\mu}, s) \\ \bar{\mu} + \rho G(z^*(\bar{\mu}, s), s) \end{bmatrix}$$  \hspace{1cm} (42)

satisfies (40).

**Proof:** The necessary condition is clear. To prove the sufficient condition, assume that

$$w^*(d_\rho(\bar{\mu}), s) = \begin{bmatrix} z^*(\bar{\mu}, s) \\ \bar{\mu} + \rho G(z^*(\bar{\mu}, s), s) \end{bmatrix}$$  \hspace{1cm} (40)

satisfies (40). The second part of (40) implies that $\mu^*(d_\rho(\bar{\mu}), s) = \bar{\mu} + \rho G(z^*(d_\rho(\bar{\mu}), s), s)$. Putting this expression in the first part of (40), one obtains that $z^*(d_\rho(\bar{\mu}), s)$ is a critical point of $L_\rho(\cdot, \bar{\mu}, s) + \iota_Z(\cdot)$.

In the sequel, a primal-dual point satisfying (40) is denoted by $w^*(d_\rho(\bar{\mu}), s)$ or $w^*(\bar{\mu}, s)$ without distinction.

As $z^\infty(\bar{\mu}_k, s_{k+1})$ is a critical point of $L_\rho(\cdot, \bar{\mu}_k, s_{k+1}) + \iota_Z(\cdot)$, one can define

$$w^\infty(d_\rho(\bar{\mu}_k), s_{k+1}) := \begin{bmatrix} z^\infty(\bar{\mu}_k, s_{k+1}) \\ \bar{\mu}_k + \rho G(z^\infty(\bar{\mu}_k, s_{k+1}), s_{k+1}) \end{bmatrix}, \hspace{1cm} (43)$$

which satisfies (40). Note that the generalised equation (40) is parametric in $s$ and $d_\rho(\cdot)$, which represents a normalised distance between a dual variable and an optimal dual variable at time $k$. Assuming that the penalty parameter $\rho$ is well-chosen, the generalised equation (40) can be proven to be strongly regular at a given solution.

**Lemma 8** (Strong regularity of (40)). There exists $\tilde{\rho} > 0$ such that for all $\rho > \tilde{\rho}$ and $k \geq 0$, (40) is strongly regular at $w^*_k = w^*(0, s_k)$.

**Proof:** In the case of a polyhedral set $Z$, this follows from the reduction procedure described in [26], the arguments developed in Proposition 2.4 in [5] and strong regularity of (14) for all $k \geq 0$. 

---

August 22, 2014 DRAFT
**Assumption 6.** The penalty parameter satisfies $\rho > \bar{\rho}$.

From the strong regularity of (40) at $w^*_k$, using Theorem 2.1 in [26], one obtains the following local Lipschitz property of a solution $w(\cdot)$ to (40).

**Lemma 9.** There exists radii $\delta_B > 0$, $r_B > 0$ and $q_B > 0$ such that for all $k \in \mathbb{N}$,

$$\forall d \in B(0, q_B), \forall s \in B(s_k, r_B), \exists! w^*(d, s) \in B(w^*_k, \delta_B),$$

$$0 \in H_\rho(w^*(d, s), d, s) + \mathcal{N}_{\mathbb{Z} \times \mathbb{R}^m}(w^*(d, s))$$

and for all $d, d' \in B(0, q_B)$ and all $s, s' \in B(s_k, r_B)$,

$$\|w^*(d, s) - w^*(d', s')\|_2 \leq \lambda_B \|H_\rho(w^*(d', s'), d, s) - H_\rho(w^*(d', s'), d', s')\|_2,$$

where $\lambda_B > 0$ is a Lipschitz constant associated with (40).

Note that, given $w \in \mathbb{Z} \times \mathbb{R}^m$, $d, d' \in \mathbb{R}^m$ and $s, s' \in \mathcal{S}$, one can write

$$H_\rho(w, d, s) - H_\rho(w, d', s') = F(w, s) - F(w, s') + \begin{bmatrix} 0 \\ d - d' \end{bmatrix},$$

which, from Assumption 3, implies the following Lemma.

**Lemma 10.** There exists $\lambda_H > 0$ such that for all $w \in \mathbb{Z} \times \mathbb{R}^m$, for all $d, d' \in \mathbb{R}^m$ and all $s, s' \in \mathcal{S}$,

$$\|H_\rho(w, d, s) - H_\rho(w, d', s')\|_2 \leq \lambda_H \left\| \begin{pmatrix} d \\ s \end{pmatrix} - \begin{pmatrix} d' \\ s' \end{pmatrix} \right\|_2.$$

**Proof:** After straightforward calculations, one obtains the Lipschitz property with

$$\lambda_H := \sqrt{\max \{ \lambda_F^2, 1 \} + \lambda_F}.$$

**C. Derivation of the contraction inequality**

In this paragraph, it is proven that under some conditions, which are made explicit in the sequel, the optimality tracking error $\|\bar{w}_k - w^*_k\|_2$ of Algorithm 1 remains within a pre-specified bound if the parameter $s_k$ varies sufficiently slowly over time.
First, note that given a sub-optimal primal-dual solution \( \bar{\mathbf{w}}_{k+1} \) and a critical point \( \mathbf{w}^*_{k+1} \),
\[
\left\| \bar{\mathbf{w}}_{k+1} - \mathbf{w}^*_{k+1} \right\|_2 \leq \left\| \bar{\mathbf{w}}_{k+1} - \mathbf{w}^\infty (d_\rho (\bar{\mu}_k), s_{k+1}) \right\|_2 + \left\| \mathbf{w}^\infty (d_\rho (\bar{\mu}_k), s_{k+1}) - \mathbf{w}^*_{k+1} \right\|_2 ,
\]
where \( \mathbf{w}^\infty (d_\rho (\bar{\mu}_k), s_{k+1}) \) has been defined in (43). The analysis then consists in bounding the two right hand side terms in (49). The first one can be upper-bounded by applying strong regularity of (40) and the second one using the convergence rate of the primal loop in Algorithm 1.

**Lemma 11.** If \( \left\| s_{k+1} - s_k \right\|_2 \) satisfies
\[
\left\| s_{k+1} - s_k \right\|_2 < \min \left\{ r_B, \frac{q_B \rho}{\lambda_A \lambda_F} \right\} ,
\]
where \( r_B \) and \( q_B \) have been defined in Lemma 9, and \( \left\| \bar{\mathbf{w}}_k - \mathbf{w}^*_{k+1} \right\|_2 < q_B \rho \), then,
\[
\left\| \mathbf{w}^\infty (d_\rho (\bar{\mu}_k), s_{k+1}) - \mathbf{w}^*_{k+1} \right\|_2 \leq \frac{\lambda_B \lambda_H}{\rho} \left( \left\| \bar{\mathbf{w}}_k - \mathbf{w}^*_{k} \right\|_2 + \lambda_A \lambda_F \left\| s_{k+1} - s_k \right\|_2 \right) .
\]

**Proof:** Note that \( \mathbf{w}^*_{k+1} \) can be rewritten \( \mathbf{w}^*_{k+1} = \mathbf{w}^* (d_\rho (\mu^*_{k+1}), s_{k+1}) \), which is a solution to (40) at \( s_{k+1} \).
\[
\left\| d_\rho (\mu^*_{k+1}) \right\|_2 = \frac{\left\| \mu^*_{k+1} - \mu_k^* \right\|_2}{\rho} \leq \frac{\lambda_F \lambda_A}{\rho} \left\| s_{k+1} - s_k \right\|_2 < q_B ,
\]
by applying Theorem 1, Assumption 3 and from hypothesis (50). Moreover,
\[
\left\| d_\rho (\bar{\mu}_k) \right\|_2 = \frac{\left\| \bar{\mu}_k - \mu_k^* \right\|_2}{\rho} \leq \frac{\left\| \bar{\mathbf{w}}_k - \mathbf{w}^*_{k} \right\|_2}{\rho} < q_B .
\]
Now, as \( \left\| s_{k+1} - s_k \right\|_2 < r_B \) one can apply Lemmas 9 and 10 to obtain
\[
\left\| \mathbf{w}^\infty (\bar{\mu}_k, s_{k+1}) - \mathbf{w}^*_{k+1} \right\|_2 \leq \lambda_B \lambda_H \left\| d_\rho (\bar{\mu}_k) - d_\rho (\mu^*_{k+1}) \right\|_2 \\
\leq \frac{\lambda_B \lambda_H}{\rho} \left( \left\| \bar{\mu}_k - \mu_k^* \right\|_2 + \left\| \mu^*_{k+1} - \mu_k^* \right\|_2 \right) \\
\leq \frac{\lambda_B \lambda_H}{\rho} \left( \left\| \bar{\mathbf{w}}_k - \mathbf{w}^*_{k} \right\|_2 + \lambda_A \lambda_F \left\| s_{k+1} - s_k \right\|_2 \right) ,
\]
by Theorem 1.

In the following Lemma, using the convergence rate estimate presented in Section IV, we derive a bound on the first summand \( \left\| \bar{\mathbf{w}}_{k+1} - \mathbf{w}^\infty (d_\rho (\bar{\mu}_k), s_{k+1}) \right\|_2 \).

**Lemma 12.** If \( \left\| s_{k+1} - s_k \right\|_2 < r_B \), \( \left\| \bar{\mathbf{w}}_k - \mathbf{w}^*_{k} \right\|_2 < q_B \rho \) and
\[
(1 + \frac{\lambda_H \lambda_B}{\rho}) \left\| \bar{\mathbf{w}}_k - \mathbf{w}^*_{k} \right\|_2 + \lambda_H \lambda_B r_B < \delta ,
\]
\[
\left\| \bar{\mathbf{w}}_{k+1} - \mathbf{w}^*_{k+1} \right\|_2 \leq \frac{\lambda_B \lambda_H}{\rho} \left( \left\| \bar{\mathbf{w}}_k - \mathbf{w}^*_{k} \right\|_2 + \lambda_A \lambda_F \left\| s_{k+1} - s_k \right\|_2 \right) .
\]
where $\delta$ has been defined in Proposition 1, then

$$
\|\bar{w}_{k+1} - w^\infty (d_\rho (\bar{\mu}_k) , s_{k+1})\|_2 \leq C (1 + \rho \lambda_g) M^{-\psi(\delta)} \left( \lambda_B \lambda_H \|s_{k+1} - s_k\|_2 \right. \\
\left. + \|\bar{w}_k - w^*_k\|_2 \left( 1 + \frac{\lambda_B \lambda_H}{\rho} \right) \right),
$$

(56)

where $\lambda_G > 0$ is the Lipschitz constant of $G(\cdot, s)$ on $Z$ (well-defined as $Z$ is bounded).

Proof: From Algorithm 1, it follows that

$$
\|\bar{w}_{k+1} - w^\infty (d_\rho (\bar{\mu}_k) , s_{k+1})\|_2 \leq \left\| \begin{pmatrix} \bar{z}_{k+1} - z^\infty (\bar{\mu}_k, s_{k+1}) \\ \rho \left( g (\bar{z}_{k+1}, s_{k+1}) - g (z^\infty (\bar{\mu}_k, s_{k+1}) , s_{k+1}) \right) \end{pmatrix} \right\|_2 \\
\leq (1 + \rho \lambda_g) \|\bar{z}_{k+1} - z^\infty (\bar{\mu}_k, s_{k+1})\|_2.
$$

(57)

In order to apply Lemma 5, one first need to show that $\bar{z}_k$ lies in the ball $B(z^\infty (\bar{\mu}_k, s_{k+1}), \delta)$, where $\delta$ is the radius involved in the KL property.

$$
\|\bar{z}_k - z^\infty (\bar{\mu}_k, s_{k+1})\|_2 \leq \|\bar{z}_k - z^* (0, s_k)\|_2 + \|z^* (0, s_k) - z^\infty (\bar{\mu}_k, s_{k+1})\|_2 \\
\leq \|\bar{w}_k - w^*_k\|_2 + \lambda_H \lambda_B \left( \|d_\rho (\bar{\mu}_k)\|_2 + \|s_{k+1} - s_k\|_2 \right) \\
\leq \left( 1 + \frac{\lambda_H \lambda_B}{\rho} \right) \|\bar{w}_k - w^*_k\|_2 + \lambda_H \lambda_B \|s_{k+1} - s_k\|_2 \\
< \delta,
$$

(58)

where the second step follows from strong regularity of (40) at $w^* (0, s_k)$ and the hypotheses mentioned above. Thus one can use the R-convergence rate estimate of Lemma 5 and apply the inequalities in (58) to obtain (56).

Gathering the results of Lemmas 11 and 12, one can state the following theorem, which upper-bounds the sub-optimality error at time $k+1$ by a linear combination of the sub-optimality error at time $k$ and the magnitude of the parameter difference.

Theorem 3 (Contraction). Given a time instant $k$, if the primal-dual error $\|\bar{w}_k - w^*_k\|_2$, the number of primal iterations $M$, the penalty parameter $\rho$ and the parameter difference $\|s_{k+1} - s_k\|_2$ satisfy

- $\|s_{k+1} - s_k\|_2 < \min \left\{ r_A, r_B, \frac{q_B \rho}{\lambda_A \lambda_F} \right\}$,
- $\|\bar{w}_k - w^*_k\|_2 < q_B \rho$,
- $\rho > \tilde{\rho}$,
\[
\left(1 + \frac{\lambda_H \lambda_B}{\rho}\right) \|\bar{w}_k - w^*_k\|_2 + \lambda_H \lambda_B \|s_{k+1} - s_k\|_2 < \delta ,
\]
then the following weak contraction is satisfied for all time instants \( k \geq 0 \):
\[
\|\bar{w}_{k+1} - w^*_k\|_2 \leq \beta_w (\rho, M) \|\bar{w}_k - w^*_k\|_2 + \beta_s (\rho, M) \|s_{k+1} - s_k\|_2 ,
\]
where
\[
\beta_w (\rho, M) := C (1 + \rho \lambda_G) \left(1 + \frac{\lambda_B \lambda_H}{\rho}\right) M^{-\psi(\hat{\theta})} + \frac{\lambda_B \lambda_H}{\rho} ,
\]
and
\[
\beta_s (\rho, M) := C (1 + \rho \lambda_G) \lambda_B \lambda_H M^{-\psi(\hat{\theta})} + \frac{\lambda_B \lambda_H \lambda_A \lambda_F}{\rho} .
\]

**Proof:** This is a direct consequence of Lemmas 11 and 12. \qed

**Remark 8.** Note that the last hypothesis (59) may be quite restrictive, since \( \|\bar{w}_k - w^*_k\|_2 \) needs to be small enough for it to be satisfied. However, in many cases the radius \( \delta \) is large (+\( \infty \) for strongly convex functions).

In order to ensure stability of the sequence of sub-optimal iterates \( \bar{w}_k \), the parameter difference \( \|s_{k+1} - s_k\|_2 \) has to be small enough and the coefficient \( \beta_w (\rho, M) \) needs to be strictly less than 1. This is clearly satisfied if the penalty parameter \( \rho \) is large enough to make \( \lambda_B \lambda_G / \rho \) small in (61). Yet the penalty parameter \( \rho \) also appears in \( 1 + \rho \lambda_G \). Hence it needs to be balanced by a large enough number of primal iterations \( M \) in order to make the first summand in (61) small. The same analysis applies to the second coefficient \( \beta_s (\rho, M) \) in order to mitigate the effect of the parameter difference \( \|s_{k+1} - s_k\|_2 \) on the sub-optimality error at \( k+1 \).

**Corollary 1** (Boundedness of the error sequence). Assume that \( \rho \) and \( M \) have been chosen so that \( \beta_w (\rho, M) \) and \( \beta_s (\rho, M) \) are strictly less than 1, and \( \rho > \tilde{\rho} \). Let \( r_w > 0 \) such that
\[
\delta - \left(1 + \frac{\lambda_H \lambda_B}{\rho}\right) r_w > 0
\]
and \( r_w < q_B \rho \). Let \( r_s > 0 \) such that
\[
r_s < \frac{(1 - \beta_w (\rho, M)) r_w}{\beta_s (\rho, M)} .
\]
If \( \| \bar{w} - w^*_0 \|_2 < r_w \) and for all \( k \geq 0 \),
\[
\| s_{k+1} - s_k \|_2 \leq \min \left\{ r_s, r_A, r_B, \frac{q_B \rho}{\lambda_A \lambda_F} \right\} ,
\]
then for all \( k \geq 0 \), the error sequence satisfies
\[
\| \bar{w}_k - w^*_k \|_2 < r_w .
\]

Proof: The proof proceeds by a straightforward induction. At \( k = 0 \), \( \| \bar{w}_0 - w^*_0 \|_2 < r_w \), by assumption. Let \( k \geq 0 \) and assume that \( \| \bar{w}_k - w^*_k \|_2 < r_w \). As \( \| s_{k+1} - s_k \|_2 < r_A \), by applying Theorem 1, there exists a unique \( w^*_{k+1} \in B(w^*_k, \delta_A) \), which satisfies (14). As \( \| s_{k+1} - s_k \|_2 \) satisfies (65), \( \| \bar{w}_k - w^*_k \|_2 < q_B \rho \), \( \rho > \bar{\rho} \) and (59) is satisfied, from the choice of \( r_w \) and \( r_s \), we have
\[
\| \bar{w}_{k+1} - w^*_{k+1} \|_2 \leq \beta_w (\rho, M) \| \bar{w}_k - w^*_k \|_2 + \beta_s (\rho, M) \| s_{k+1} - s_k \|_2
\]
\[
\leq \beta_w (\rho, M) r_w + \beta_s (\rho, M) \| s_{k+1} - s_k \|_2
\]
\[
\leq r_w ,
\]
as \( \| s_{k+1} - s_k \|_2 \leq r_s < (1 - \beta_w(\rho, M)) r_w / \beta_s(\rho, M) \). Note from the choice of \( r_w \) and \( r_s \), the condition (59) guaranteeing the weak contraction (60) is also recursively satisfied.

D. Improved contraction via continuation

In Algorithm 1, only one dual update is performed to move from state \( s_k \) to state \( s_{k+1} \), in contrast to standard augmented Lagrangian techniques where the Lagrange multiplier \( \mu \) and the penalty parameter \( \rho \) are updated after every sequence of primal iterations. Intuitively, one would expect that applying several dual updates instead of just one, drives the suboptimal solution \( \bar{w}_{k+1} \) closer to the optimal one \( w^*_{k+1} \), thus enhancing the tracking performance over time. However, as the number of primal iterations \( M \) is fixed a priori, it is not obvious at all why this would happen, as primal iterations generally need to become more accurate when the dual variable moves closer to optimality. Therefore, we resort to an homotopy-based mechanism to fully exploit property (60). Continuation techniques [1], in which the optimal solution is parameterised along an homotopy path from the previous state \( s_k \) to the current one \( s_{k+1} \), have been successfully applied for solving online convex quadratic programs in the QPOASES package [18].
The state parameter $s$ can be seen as an extra degree of freedom in Algorithm 1, which can be modified along the iterations. More precisely, instead of carrying out a sequence of alternating proximal gradient steps to find a critical point of $L(\cdot, \bar{\mu}_k, s_{k+1}) + \delta_Z(\cdot)$ directly at the parameter $s_{k+1}$, one moves from $s_k$ towards $s_{k+1}$ step by step, each step corresponding to a dual update and a sequence of alternating proximal gradients. The proposed approach can be seen as a form of ‘tracking in the tracking’. More precisely, one defines a finite sequence $\{s^j_k\}$ of $D$ state-parameters along the homotopy path $\{(1 - \tau) s_k + \tau s_{k+1} \mid \tau \in [0, 1]\}$ by

$$s^j_k := \left(1 - \frac{j}{D}\right) s_k + \frac{j}{D} s_{k+1}, \quad j \in \{0, \ldots, D\},$$

where $D \geq 2$. This modification results in Algorithm 3 below. At every step $j$, an homotopy update is first carried out. A sequence of proximal minimisation is then applied given the current state $s$ and multiplier $\mu$, which is updated at the end of step $j$. In a sense, Algorithm 3 consists in repeatedly applying Algorithm 1 on an artificial dynamics determined by the homotopy steps.

The reason for introducing Algorithm 3 is that it allows for a stronger contraction effect on the sub-optimality gap $\|\bar{w}_{k+1} - w^*_{k+1}\|_2$ than Algorithm 1, as formalised by the following Theorem.

**Lemma 13** (Optimality along the homotopy path). Given a time instant $k \geq 0$, for all $j \in \{1, \ldots, D\}$, there exists a unique primal-dual variable $w^*\left(s^j_k\right) \in B\left(z^*_k, r_A\right)$ satisfying

$$0 \in F\left(w^*\left(s^j_k\right), s^j_k\right) + N_{\mathbb{Z} \times \mathbb{R}^m}\left(w^*\left(s^j_k\right)\right).$$

**Proof:** This comes directly from the strong regularity of (14), Assumption 5 and $\|s^j_k - s_k\|_2 \leq \|s_{k+1} - s_k\|_2$ for all $j \in \{1, \ldots, D\}$.

**Remark 9.** Note that the NMPC program (8) at state $s^j_k$, $j \in \{1, \ldots, D\}$, is feasible, by strong regularity of (14) at $w^*\left(s^0_k\right)$, since $\|s^j_k - s_k\|_2 < r_A$. However, in general, for an arbitrarily large parameter difference $\|s_{k+1} - s_k\|_2$, this is not true, as the feasible set of the NMPC controller associated with (8) is not convex.

**Assumption 7.** For all time instants $k$ and $j \in \{1, \ldots, D\}$, the generalised equation (14) is strongly regular at its solution $w^*\left(s^j_k\right)$.

**Theorem 4** (Improved contraction via continuation). Assume that $\rho > \bar{\rho}$ and that $\rho$ and $M$ have been chosen so that $\beta_w(\rho, M), \beta_s(\rho, M) < 1$. Given a time instant $k \geq 0$, if $\|\bar{w}_k - w^*_k\|_2 < r_w$, August 22, 2014 DRAFT
Algorithm 3 Homotopy-based optimality tracking splitting algorithm

**Input:** Suboptimal primal-dual solution $(\bar{z}^T_k, \bar{\mu}_k)^T$, state parameters $s_k$ and $s_{k+1}$.

$s \leftarrow s_k$, $\mu \leftarrow \bar{\mu}_k$, $z_{\text{wms}} \leftarrow \bar{z}_k$

**Continuation loop:**

for $j = 1 \ldots D$ do

$s \leftarrow s + \frac{s_{k+1} - s_k}{D}$

**Primal loop:**

$z^{(0)} \leftarrow z_{\text{wms}}$

$c_i^{(-1)} \leftarrow c_i^{\text{wms}}$ for $i \in \{1, \ldots, P\}$

for $l = 0 \ldots M - 1$ do

for $i = 1 \ldots P$ do

$(z_i^{(l+1)}, c_i^{(l)}), c_i^{(l-1)}, \mu, \rho, s)$

end for

end for

$z_{\text{wms}} \leftarrow z^{(M)}$

**Dual update:** $\mu \leftarrow \mu + \rho G(z^{(M)}, s)$

end for

$\bar{z}_{k+1} \leftarrow z_{\text{wms}}; \bar{\mu}_{k+1} \leftarrow \mu$

where $r_w$ satisfies the assumptions of Corollary 1, and $\|s_{k+1} - s_k\|_2$ satisfies (65), then the primal-dual sub-optimal variable $\bar{w}_{k+1}$ yielded by Algorithm 3 satisfies the following inequality

$$
\left\| \bar{w}_{k+1} - \bar{w}^*_{k+1} \right\|_2 \leq \beta_w^D (\rho, M) \left\| \bar{w}_k - \bar{w}_k^* \right\|_2 + \beta_s (\rho, M) \sum_{i=0}^{D-1} \beta_i^w (\rho, M) \left\| s_{k+1} - s_k \right\|_2 .
$$

(70)

**Proof:** For all $j \in \{1, \ldots, D\}$, define $\bar{\mu}_k^j := \bar{\mu}_k + \rho G \left( \bar{z}_k^j, \bar{s}_k^j \right)$ with $\bar{\mu}_0^j := \bar{\mu}_k$ and where $\bar{z}_k^j$ is obtained after $M$ alternating proximal gradient steps applied to $L_{\rho} \left( \cdot, \bar{\mu}_k^j, \bar{s}_k^j \right) + \delta_Z (\cdot)$. One can thus define a sub-optimal primal-dual variable $\bar{w}_k^j := \left( (\bar{z}_k^j)^T, (\bar{\mu}_k^j)^T \right)^T$ for the homotopy state $s_k^j$. By applying Corollary 1, one obtains that for all $j \in \{0, \ldots, D-1\}$, $\left\| \bar{w}_k^j - \bar{w}^* (s_k^j) \right\|_2 < r_w < q_B \rho$, since $\left\| s_{k+1}^j - s_k^j \right\|_2 = \left\| s_{k+1} - s_k \right\|_2 / D < \min \left\{ r_s, r_A, r_B, \frac{q_B \rho}{\lambda_A \lambda F} \right\}$. It can also be readily shown that for all $j \in \{0, \ldots, D-1\}$,

$$
\left( 1 + \frac{\lambda_H \lambda_B}{\rho} \right) \left\| \bar{w}_k^j - \bar{w}^* (s_k^j) \right\|_2 + \lambda_H \lambda_B \left\| s_{k+1}^j - s_k^j \right\|_2 < \delta .
$$

(71)
Subsequently, one can apply the same reasoning as for proving Theorem 3, and get that for all \( j \in \{0, \ldots, D - 1\} \),
\[
\|\bar{w}_k^{j+1} - w^* (s_k^{j+1})\|_2 \leq \beta_w (\rho, M) \|\bar{w}_k^j - w^* (s_k^j)\|_2 + \beta_s (\rho, M) \|s_k^{j+1} - s_k^j\|_2 .
\] (72)

By iterating inequality (72) from \( j = 0 \) to \( D - 1 \), we obtain
\[
\|\bar{w}_{k+1} - w^*_{k+1}\|_2 \leq \beta_w (\rho, M) \|\bar{w}_k^{D-1} - w^* (s_k^{D-1})\|_2 + \frac{\beta_s (\rho, M)}{D} \|s_{k+1} - s_k\|_2 
\]
\[
\leq \ldots 
\leq \beta_w^D (\rho, M) \|\bar{w}_k^0 - w^* (s_k^0)\|_2 + \beta_s (\rho, M) \frac{\sum_{j=0}^{D-1} \beta_w^j (\rho, M)}{D} \|s_{k+1} - s_k\|_2 ,
\]
which is exactly inequality (70).

As \( \beta_w (\rho, M) < 1 \), \( \beta_s (\rho, M) < 1 \) and \( D \geq 2 \), it follows that \( \beta_w^D (\rho, M) < \beta_w (\rho, M) \) and \( \beta_s (\rho, M) \frac{\sum_{j=0}^{D-1} \beta_w^j (\rho, M)}{D} < \beta_s (\rho, M) \), which implies that the contraction (70) is stronger than (60). In practice, the coefficients \( \beta_w (\rho, M) \) and \( \beta_s (\rho, M) \) in (60) can be reduced by an appropriate tuning of the penalty \( \rho \) and the number of primal proximal steps \( M \). Yet this approach is limited, as previously discussed in Paragraph V-C. Therefore, Algorithm 3 provides a more efficient and systematic way of improving the optimality tracking performance. Superiority of Algorithm 3 over Algorithm 1 is demonstrated on a numerical example in Section VII.

VI. COMPUTATIONAL CONSIDERATIONS

By making use of partial penalisation, Algorithm 1 allows for a more general problem formulation than [32], where the primal QP sub-problem is assumed to have non-negativity constraints only. In contrast, our framework can handle any convex constraint set \( Z_i \) for which the proximal operator (1) is cheap to compute. This happens when \( Z_i \) is a ball, an ellipsoid, a box, the non-negative orthant or even second-order conic constraints and the semi-definite cone.

Remark 10. For many non-convex sets, such as spheres or mixed-integer sets, the proximal operator (1) can be obtained in closed-form. However, the analysis of Section V does not readily extend, as Robinson’s strong regularity is defined for closed convex sets [26].

Remark 11. In a distributed framework, any convex set \( Z_i \) could be handled in Algorithm 1, as every agents would then end up solving a local convex NLP.
Algorithm 1 can be further refined by introducing local copies of the variables. Considering the NLP

\[
\begin{align*}
\text{minimise} & \quad J(z_1, \ldots, z_P) \\
\text{s.t.} & \quad G(z_1, \ldots, z_P) = 0 \\
& \quad z_1 \in \mathcal{Z}_1, \ldots, z_P \in \mathcal{Z}_P,
\end{align*}
\]

variables \(y_i\) can be incorporated in the equality constraints, resulting in

\[
\begin{align*}
\text{minimise} & \quad J(y_1, \ldots, y_P) \\
\text{s.t.} & \quad G(y_1, \ldots, y_P) = 0 \\
& \quad y_i - z_i = 0 \quad \forall i \in \{1, \ldots, P\} \\
& \quad z_1 \in \mathcal{Z}_1, \ldots, z_P \in \mathcal{Z}_P.
\end{align*}
\]

Subsequently, at iteration \(l + 1\), some of the alternations are given by

\[
\begin{align*}
\text{minimise} & \quad \nu_i^T \left( y_i^{(l+1)} - z_i \right) + \frac{\rho}{2} \left\| y_i^{(l+1)} - z_i \right\|^2 + \frac{\alpha_i}{2} \left\| z_i - z_i^{(l)} \right\|^2,
\end{align*}
\]

where \(\nu_i\) is a dual variable associated with the equality constraint \(y_i - z_i = 0\). This step can be rewritten

\[
\begin{align*}
\text{minimise} & \quad \left\| z_i - \frac{1}{\alpha_i + \rho} \left( \alpha_i z_i^{(l)} + \rho y_i^{(l+1)} + \nu_i \right) \right\|^2,
\end{align*}
\]

which corresponds to projecting

\[
\frac{1}{\alpha_i + \rho} \left( \alpha_i z_i^{(l)} + \rho y_i^{(l+1)} + \nu_i \right)
\]

onto \(\mathcal{Z}_i\). This type of an approach is useful if the minimisation over the \(y_i\) variables is tractable, for instance when \(J(\cdot)\) is multi-convex and \(G(\cdot)\) is multilinear, and the projection onto \(\mathcal{Z}_i\) is cheap to compute.

**VII. Numerical Examples**

Algorithms 1 and 3 are tested on two nonlinear systems, a DC motor (centralised) in paragraph VII-A and a formation of three unicycles (distributed) in paragraph VII-B. The effect of the parameters \(\rho\) and \(\Delta t\) is analysed, assuming that a fixed number of proximal operators can be computed per second. In particular, it is shown that the theoretical results proven in Section...
V are able to predict the practical behaviour of the combined system-optimiser dynamics quite well, and that tuning the optimiser’s step-size $\rho$ and the system’s step-size $\Delta t$ has a strong effect on the closed-loop trajectories.

A. DC motor

The first example is a DC motor with continuous-time bilinear dynamics

$$\dot{x} = Ax + Bu + c, \quad (78)$$

where

$$A = \begin{pmatrix} \frac{-R_a}{L_a} & 0 \\ 0 & \frac{-B}{J} \end{pmatrix}, \quad B = \begin{pmatrix} 0 & \frac{-k_m}{L_a} \\ \frac{k_m}{J} & 0 \end{pmatrix}, \quad c = \begin{pmatrix} \frac{u_a}{L_a} \\ -\frac{\tau_l}{J} \end{pmatrix}, \quad (79)$$

and the parameters are borrowed from the experimental identification presented in [14]:

$$L_a = 0.307 \, \text{H}, \quad R_a = 12.548 \, \Omega, \quad k_m = 0.22567 \, \text{Nm/A}^2,$$
$$J = 0.00385 \, \text{Nm.sec}^2, \quad B = 0.00783 \, \text{Nm.sec},$$
$$\tau_l = 1.47 \, \text{Nm}, \quad u_a = 60 \, \text{V}. \quad (80)$$

The first component of the state variable $x_1$ is the armature current, while the second component $x_2$ is the angular speed. The control input $u$ is the field current of the machine. The control objective is to make the angular speed track a piecewise constant reference $x^\text{ref}_r = \pm 2 \, \text{rad/sec}$, while satisfying the following state and input constraints:

$$\underline{x} = \begin{pmatrix} -2 \, \text{A} \\ -8 \, \text{rad/sec} \end{pmatrix}, \quad \overline{x} = \begin{pmatrix} 5 \, \text{A} \\ 1.5 \, \text{rad/sec} \end{pmatrix},$$
$$\underline{u} = 1.27 \, \text{A}, \quad \overline{u} = 1.4 \, \text{A}. \quad (81)$$

The continuous-time NMPC problem for reference tracking is discretised at a given sampling period $\Delta t$ using an explicit Euler method, which results in a bilinear NLP. Although the consistency of the explicit Euler integrator is 1, only the first control input is applied to the real system, implying that the prediction error with respect to the continuous-time dynamics is small. For simulating the closed-loop system under the computed NMPC control law, the MATLAB integrator ode45 is used with the sampling period $\Delta t$. The prediction horizon is fixed at 30 samples. This is a key requirement for the analysis that follows, as explained later.
Fig. 1. Angular speed against time for increasing sampling periods $\Delta t$ and a fixed computational power $2 \cdot 10^3 \text{ prox/sec}$: 0.004 sec (top), 0.018 sec (middle) and 0.04 sec (bottom). The sub-optimal trajectory obtained with Algorithm 1 is plotted in dashed red, while the full NMPC trajectory obtained using IPOPT (for the same $\Delta t$) is in blue.

In general, the computational power of an embedded computing platform is quite limited, meaning that the total number of proximal steps that can be computed within one second by Algorithms 1 and 3 is fixed and finite. Later on, we refer to this number as the computational power, expressed in $\text{prox/sec}$. The results plotted in Figs. 1, 2, 3 and 4 are obtained for a compu-
Fig. 2. Angular speed against time for increasing penalty parameters $\rho$ and a fixed computation power $2 \cdot 10^3$ prox/sec: 20 (top), 100 (middle) and 1000 (bottom). The sub-optimal trajectory obtained with Algorithm 1 is plotted in dashed red, while the full NMPC trajectory obtained using IPOPT (for the same $\Delta t$) is in blue.

tational power of $2 \cdot 10^3$ prox/sec. In Fig. 1, it clearly appears that a better tracking performance is obtained for $\Delta t = 0.018$ sec, compared to a lower sampling period ($\Delta t = 0.004$ sec) or a larger sampling period ($\Delta t = 0.04$ sec). The effect of the system’s step-size $\Delta t$ on the performance of Algorithm 1 given a fixed computational power is demonstrated more clearly in Fig. 5.
Remark 12. For the purpose of this analysis, the computation time of the dual updates in Algorithms 1 and 3 is neglected.

Another key parameter is the penalty coefficient $\rho$, which can also be interpreted as a step-size for the optimiser. In order to demonstrate the effect of $\rho$ on the efficacy of our optimality tracking splitting scheme, the sampling period $\Delta t$ is fixed at $0.018$ sec given a computational power of $2 \cdot 10^3$ prox/sec, which implies that the total number of primal iterations is $M = 36$, and $\rho$ is made vary within $\{20, 100, 1 \cdot 10^3\}$. Figure 2 shows that a better tracking performance is obtained with $\rho = 100$ than with $\rho = 20$ or $\rho = 1 \cdot 10^3$. This can be deduced from the expression of the coefficient $\beta_w(\rho, M)$ in Eq. (61), as explained in Paragraph V-C. The optimal choice of the penalty parameter is known to be critical to the convergence speed of ADMM, which is very similar to our optimality tracking splitting scheme. To our knowledge, this effect has only been observed for ADMM-type techniques when dealing with convex programs. When solving non-convex programs using augmented Lagrangian, it is commonly admitted that $\rho$ should be chosen large enough in order to ensure (locally) positive definiteness of the hessian of the augmented Lagrangian. Taking $\rho$ too large is known to result in ill-conditioning. For Algorithm 1, which is essentially a first-order method, the analysis is different, as $\rho$ does not affect the algorithm at the level of linear algebra, but does impact the contraction of the primal-dual sequence, and thus the convergence speed over time, or tracking performance. Thus our study provides a novel interpretation of the choice of $\rho$ via a parametric analysis in a non-convex framework. The effect of the optimiser step-size $\rho$ on the closed-loop performance fully appears in Fig. 6.

Satisfaction of the KKT conditions of the parametric augmented Lagrangian problem

$$\minimize_{z \in B(\bar{z}, \bar{z})} L_{\rho}(z, \bar{\mu}_k, s_{k+1})$$

is measured along the closed-loop trajectory by computing

$$\omega_k := \|\pi_{B(\bar{z}, \bar{z})}(\bar{z}(\bar{\mu}_{k-1}, s_k) - \nabla L_{\rho}(\bar{z}(\bar{\mu}_{k-1}, s_k)), \bar{\mu}_{k-1}, s_k)) - \bar{z}(\bar{\mu}_{k-1}, s_k)\|_2,$$

which is plotted in Fig. 3. Over time, convergence towards low criticality values is faster for $\Delta t = 0.18$ sec, than for shorter sampling period ($\Delta t = 0.004$ sec) or larger sampling period ($\Delta t = 0.04$ sec). The same effect can be observed for the feasibility of the nonlinear equality constraints $G(\cdot, s_k)$, as pictured in Fig. 4.
Fig. 3. Optimality of bound constrained augmented Lagrangian program for different sampling periods $\Delta t$ and a fixed computation power $2 \cdot 10^3$ $\text{prox/sec}$: 0.004 sec (black), 0.018 sec (red) and 0.04 sec (blue).

Fig. 4. Norm of equality constraints $\|G(\bar{z}_k, s_k)\|_2$ for different sampling periods $\Delta t$ and a fixed computation power $2 \cdot 10^3$ $\text{prox/sec}$: 0.004 sec (black), 0.018 sec (red) and 0.04 sec (blue).
From the results presented in Figures 1, 2, 3 and 4, one may conclude that sampling faster does not necessarily result in better performance of Algorithm 1. This surprising behaviour is confirmed by Figure 5. For every computational power within \( \{1 \cdot 10^3, 2 \cdot 10^3, 3 \cdot 10^3, 4 \cdot 10^3\} \), the sampling period is made vary from \( \Delta t = 2 \cdot 10^{-3} \) sec to \( \Delta t = 4 \cdot 10^{-2} \) sec. The tracking performance is assessed by computed the normalised \( L^2 \)-norm of the difference between the full-NMPC output trajectory obtained with \texttt{IPOPT} [31] and the output signal obtained with Algorithm 1 (at the same sampling period), on a fixed time interval between 2 sec and 4 sec. More precisely, the optimality tracking error is defined by

\[
E := \sqrt{\frac{1}{N_s} \sum_{k=1}^{N_s} (y^*_k - \bar{y}_k)^2},
\]

where \( \{y^*_k\} \) is the system output signal obtained with \texttt{IPOPT}, \( \{\bar{y}_k\} \) is the system output signal obtained with Algorithm 1 (for the same \( \Delta t \)) and \( N_s \) is the number of time samples. For a fast sampling, the error \( E \) appears to be quite large (\( 1 \cdot 10^0 \)), as the warm-starting point is close to the optimal solution but only few primal proxies can be evaluated, resulting in little improvement of the initial guess in terms of optimality. This effect can even be justified further by Theorem 3: as the number of primal iterations \( M \) is fixed by the sampling period, the term \( M^{-\psi(\hat{\theta})} \) in the expression of \( \beta_w(\rho, M) \) and \( \beta_s(\rho, M) \) is not small enough to dampen the effect of the term \( 1 + \rho \lambda G \), and thus the contraction (60) becomes looser, thus degrading the closed-loop performance. As the sampling becomes slower, more primal proximal iterations can be carried out and subsequently, the error \( E \) is reduced. The same reasoning as before on \( \beta_w(\rho, M) \) and \( \beta_s(\rho, M) \) can be made. However, if the sampling frequency \( 1/\Delta t \) is too low, the initial guess is very far from the optimal point, to the point that Assumption 5 may not be satisfied anymore, hence the error increases again. Thus, at every computational power, an optimal sampling period is obtained. As the computation power increases, the optimal \( \Delta t \) appears to decrease and the associated optimality tracking error \( E \) drops.

**Remark 13.** Note that we compare the behaviour of our parametric optimisation algorithm on NLPs of fixed dimension, no matter what the sampling period is, as the number of prediction samples has been fixed. This means that the prediction time changes as the sampling period varies, which may have an effect on the closed-loop behaviour. However, it is important to remember that the error \( E \) is measured with respect to the closed-loop trajectory under the optimal full-NMPC control law computed at the same sampling period.
An interesting aspect of the non-convex splitting Algorithm 1 is that the step-size $\rho$ has an effect on the closed-loop behaviour of the nonlinear dynamics, as shown in Fig. 6. Given fixed...
sampling period and computational power, the tracking performance can be improved by tuning the optimiser step-size $\rho$. In a sense, $\rho$ can now be interpreted as a tuning parameter for the NMPC controller. In particular, for a fixed number of primal iterations $M$, choosing $\rho$ too large makes the numerical value of the contraction coefficients $\beta_w(\rho, M)$ and $\beta_s(\rho, M)$ blow up, subsequently degrading the tracking performance.

![Graph showing the evolution of the optimality tracking error $E$ against sampling period $\Delta t$.](image)

**Fig. 7.** Evolution of the optimality tracking error $E$ against sampling period $\Delta t$. Algorithm 1 for $3 \cdot 10^3 \text{ prox/sec}$ in black, for $4 \cdot 10^3 \text{ prox/sec}$ in blue. Algorithm 3 with 3 homotopy steps for $3 \cdot 10^3 \text{ prox/sec}$ in dashed red, with 4 homotopy steps for $4 \cdot 10^3 \text{ prox/sec}$ in red.

From the arguments developed in paragraph V-D of Section V, one can expect Algorithm 3 to track the time-dependent optima more closely than Algorithm 1. This is confirmed by Fig. 7.

### B. Collaborative tracking of unicycles

The second example is a collaborative tracking problem based on NMPC. Three unicycles are controlled so that a leader follows a predefined path, while two followers maintain a fixed formation. This control objective can be translated into the cost function of an NMPC problem,
which is then written

\[
\int_0^T \left( \left\| x^{(1)}(t) - x^r(t) \right\|_{Q_1}^2 + \left\| u^{(1)}(t) \right\|_{R_1}^2 + \left\| u^{(2)}(t) \right\|_{R_2}^2 + \left\| u^{(3)}(t) \right\|_{R_3}^2 
+ \left\| x^{(1)}(t) - x^{(2)}(t) - d_{1,2} \right\|_{Q_{1,2}}^2 + \left\| x^{(1)}(t) - x^{(3)}(t) - d_{1,3} \right\|_{Q_{1,3}}^2 \right) dt ,
\]

(85)

where \( Q_1, Q_{1,2}, Q_{1,3}, R_1, R_2, R_3 \) are positive definite matrices, \( d_{1,2}, d_{1,3} \) are vectors that define the formation between unicycles 1, 2 and 3 and \( x^r(\cdot) \) is a reference path. All agents 1, 2 and 3 follow the standard unicycle dynamics

\[
\begin{align*}
\dot{x}_1 &= u_1 \cos x_3 \\
\dot{x}_2 &= u_1 \sin x_3 \\
\dot{x}_3 &= u_2
\end{align*}
\]

(86)

\[
\begin{array}{c}
\text{Reference} \\
\text{IPOPT} \\
\text{AL (param.)}
\end{array}
\]

Fig. 8. Trajectories of the three-unicycles formation for 300 m/s, \( \Delta t = 0.35 \) sec and \( \rho = 2 \cdot 10^3 \).

The continuous-time NMPC problem is discretised using a Runge-Kutta integrator of order 4 [19], while the closed-loop system is simulated with the MATLAB varying step-size integrator \texttt{ode45}. In the resulting finite-dimensional NLP, two cost coupling terms appear between agents 1 and 2, and agents 1 and 3. This can be addressed by the splitting Algorithm 1. Moreover, the whole procedure then consists in a sequence of proximal alternating steps between agent 1 and the group...
\{2, 3\}, which can compute their proximal descents in parallel without requiring any communication. For this particular NLP with cost-couplings, the dual updates can be performed in parallel.

\[
\epsilon_{1,2} := \left\| x^{(1)} - x^{(2)} - d_{1,2} \right\|_2
\]

is plotted in Fig. 9. At every reference change, the error rises, but decreases again as the tracking converges. The performance could be further improved by tuning the penalty \( \rho \) or performing a few homotopy steps as in Algorithm 3.

Results of the collaborative tracking NMPC are presented in Figures 8 and 9. The number of iterations per second has been fixed at 300 \text{prox/sec} and the sampling period set to \( \Delta t = 0.35 \) sec, which effectively results in \( M = 105 \). The penalty parameter was \( \rho = 2 \cdot 10^3 \). The formation-keeping NMPC has been first simulated with the unicycles in closed-loop with the full-NMPC control law, computed using IPOPT with accuracy \( 1 \cdot 10^{-7} \), which is purely centralised, hence not very interesting from a practical point of view, in this particular case. The full-NMPC trajectory is plotted in black in Fig. 8, while the one obtained using Algorithm 1 is represented in blue. The closed-loop formation error

Fig. 9. Evolution of the formation error between unicycles 1 and 2 for Algorithm 1 (blue), compared with the formation error obtained with the full NMPC (IPOPT, black).
VIII. CONCLUSION

An novel non-convex splitting Algorithm for solving NMPC programs in a real-time framework has been presented. Contraction of the primal-dual sequence has been proven using regularity of generalised equations and recent results on non-convex descent methods. It has been shown that the proposed Algorithm can be further improved by applying a continuation technique. Finally, the proposed approach has been analysed in details on two numerical examples.

REFERENCES

[1] E.L. Allgower and K. Georg. Introduction to numerical continuation methods. SIAM, 1990.
[2] H. Attouch and J. Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. Mathematical Programming, 116:5–16, 2009.
[3] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimisation and projection methods for non-convex problems: an approach based on the Kurdyka-Lojasiewicz inequality. Mathematics of Operations Research, 35:438–457, 2010.
[4] H. Attouch, J. Bolte, and B.F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting and regularised Gauss-Seidel methods. Mathematical Programming, 137:91–129, 2013.
[5] D.P. Bertsekas. Constrained optimization and Lagrange multiplier methods. Athena Scientific, 1982.
[6] D.P. Bertsekas and J.N. Tsitsiklis. Parallel and distributed computation: numerical methods. Athena Scientific, 1997.
[7] J. Bolte, A. Daniilidis, and A. Lewis. The Lojasiewicz inequality for nonsmooth sub-analytic functions with applications to subgradient dynamical systems. SIAM Journal on Optimization, 17:1205–1223, 2007.
[8] J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearised minimisation for nonconvex and nonsmooth problems. Mathematical Programming, 2013.
[9] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Maching Learning, 3(1):1–122, 2010.
[10] A.R. Conn, N.I.M. Gould, A. Sartenaer, and P.L. Toint. Convergence properties of an augmented Lagrangian algorithm for optimization with a combination of general equality and linear constraints. SIAM Journal on Optimization, 6(3):674–703, August 1996.
[11] A.R. Conn, N.I.M Gould, and P.L. Toint. Testing a class of methods for solving minimisation problems with simple bounds on the variables. Mathematics of Computation, 50(182):399–430, 1988.
[12] A.R. Conn, N.I.M. Gould, and P.L. Toint. A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. SIAM Journal on Numerical Analysis, 28:545–572, 1991.
[13] D. D’Acunto and K. Kurdyka. Explicit bounds for the Lojasiewicz exponent in the gradient inequality for polynomials. Annales Polonici Mathematici, 87, 2005.
[14] S. Daniel-Berhe and H. Unbehauen. Experimental physical parameter estimation of a thyristor driven DC-motor using the HMF-method. Control Engineering Practice, 6:615–626, 1998.
[15] M. Diehl, H.G. Bock, and J.P. Schlöder. A real-time iteration scheme for nonlinear optimization in optimal feedback control. SIAM Journal on Control and Optimization, 43(5):1714–1736, 2005.
[16] Q. Tran Dinh, I. Necoara, and M. Diehl. A dual decomposition algorithm for separable nonconvex optimization using the penalty framework. In Proceedings of the 52nd Conference on Decision and Control, 2013.

[17] A.L. Dontchev and R.T. Rockafellar. Characterizations of strong regularity for variational inequalities over polyhedral convex sets. SIAM Journal on Optimization, 6(4):1087–1105, 1996.

[18] 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, 18:816–830, 2008.

[19] E. Hairer and G. Wanner. Solving ordinary differential equations I. Springer, 2008.

[20] A. Hamdi and S.K. Mishra. Decomposition methods based on augmented Lagrangian: a survey. In Topics in nonconvex optimization. Mishra, S.K., 2011.

[21] F. Lin, M. Fardad, and M.R. Jovanovic. Design of optimal sparse feedback gains via the alternating direction method of multipliers. IEEE Transactions on Automatic Control, 58(9), 2013.

[22] S. Lojasiewicz. Une propriété topologique des sous-ensembles analytiques réels. In Les Équations aux Dérivées Partielles, pages 87–89. Éditions du CNRS, 1963.

[23] I. Necoara, C. Savorgnan, Q. Tran Dinh, J. Suykens, and M. Diehl. Distributed nonlinear optimal control using sequential convex programming and smoothing techniques. In Proceedings of the 48th Conference on Decision and Control, 2009.

[24] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM Journal on Optimization, 22(2):341–362, 2012.

[25] M.J.D. Powell. On search directions for minimisation algorithms. Mathematical Programming, 4:193–201, 1973.

[26] Stephen M. Robinson. Strongly regular generalised equations. Mathematics of Operations Research, 5(1):43–62, 1980.

[27] R.T. Rockafellar. Augmented Lagrangian multiplier functions and duality in non-convex programming. SIAM Journal on Control and Optimization, 12(2):268–285, 1974.

[28] R.T. Rockafellar and R. J.-B. Wets. Variational Analysis. Springer, 2009.

[29] Q. Tran Dinh, C. Savorgnan, and M. Diehl. Adjoint-based predictor-corrector sequential convex programming for parametric nonlinear optimization. SIAM Journal on Optimization, 22(4):1258–1284, 2012.

[30] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. Journal of Optimization Theory and Applications, 109(3):475–494, 2001.

[31] A. Waechter and L.T. Biegler. On the implementation of a primal-dual interior point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming, 106(1):25–57, 2006.

[32] V.M. Zavala and M. Anitescu. Real-time nonlinear optimization as a generalised equation. SIAM Journal on Control and Optimization, 48(8):5444–5467, 2010.

[33] V.M. Zavala and M. Anitescu. Scalable nonlinear programming via exact differentiable penalty functions and trust-region Newton methods. SIAM Journal on Optimization, 24(1):528–558, 2014.

[34] V.M. Zavala and L.T. Biegler. The advanced-step NMPC controller: optimality, stability and robustness. Automatica, 45(1):86–93, 2009.