2014 American Control Conference (ACC)
June 4-6, 2014. Portland, Oregon, USA

An Improved Distributed Dual Newton-CG Method for Convex Quadratic Programming Problems

Attila Kozma[1], Emil Klintberg[2], Sebastien Gros[2] and Moritz Diehl[1]

Abstract— **This paper considers the problem of solving Quadratic Programs (QP) arising in the context of distributed optimization and optimal control. A dual decomposition approach is used, where the QP subproblems are solved locally, while the constraints coupling the different subsystems in the time and space domains are enforced by performing a distributed non-smooth Newton iteration on the dual variables. The iterative linear algebra method Conjugate Gradient (CG) is used to compute the dual Newton step. In this context, it has been observed that the dual Hessian can be singular when a poor initial guess for the dual variables is used, hence leading to a failure of the linear algebra. This paper studies this effect and proposes a constraint relaxation strategy to address the problem. It is both formally and experimentally shown that the relaxation prevents the dual Hessian singularity. Moreover, numerical experiments suggest that the proposed relaxation improves significantly the convergence of the Distributed Dual Newton-CG.**

I. Introduction

Distributed Quadratic Programming (QP) problems arise in many applications of optimization and optimal control. They arise when Nonlinear Programs with decomposable cost functions are solved via Sequential Quadratic Programming (SQP) type methods [1]; or when Model Predictive Control (MPC) is applied to a set of sparsely interconnected subsystems, where the control problem is ultimately formulated as a QP whose structure reflects the distributed nature of the problem; and similarly in Nonlinear Model Predictive Control (NMPC) once the dynamics of the subsystems are discretized via e.g. multiple shooting methods [2], [3]. In this context, the different interconnected subsystems aim at reaching an overall optimal cost while respecting their local operational limitations.

In this paper, we will consider generic decomposable convex QP problem of the form

$$\min_{x} \sum_{k=1}^{N} \frac{1}{2} x_k^T H_k x_k + c_k^T x_k \tag{1a}$$

$$\text{s.t. } A_{i,j} x_i = B_{i,j} x_j \quad (i,j) \in E \tag{1b}$$

$$x_l \in \mathcal{X}_l \ , \quad l = 1, \dots, n, \tag{1c}$$

where, for all $k = 1, \dots, N$, $x_k \in \mathbb{R}^{n_k}$ are the optimization variables with $x^T = [x_1^T, \dots, x_N^T] \in \mathbb{R}^n$. Furthermore, $0 \prec H_k \in \mathbb{R}^{n_k \times n_k}$, $c_k \in \mathbb{R}^{n_k}$, $\mathcal{G} = (V, E)$ denotes a directed graph describing the topology of the connecting subproblems. Matrices $A_{i,j} \in \mathbb{R}^{v_{i,j} \times n_i}$ and $B_{i,j} \in \mathbb{R}^{v_{i,j} \times n_j}$ yield the coupling constraints between the subproblems i and j, respectively, for all $(i,j) \in E$.

Set $\mathcal{X}_k$ denotes a polyhedral set representing the local inequality constraints, i.e. we define $\mathcal{X}_k := \{y \in \mathbb{R}^{n_k} | D_k y \le e_k\}$ with $D_k \in \mathbb{R}^{w_k \times n_k}$. In the development proposed in this paper, for the sake of simplicity but without loss of generality, we do not consider equality constraints. Indeed, since equality constraints can always be eliminated from the problem via a null-space reformulation, the arguments presented here remain valid in the case equality constraints are considered.

Several centralized methods have been proposed to solve (1), e.g. [4], [5], by exploiting the inherent sparsity structure of the problem yielded by the distributed nature of the problem. However, these methods are of limited use once the problem data does not fit into the shared memory. Moreover, it is often not desirable or highly impractical to centralise the data of the problem. This is the case when e.g. the subsystems are geographically distributed and heavy long-distance communications are to be avoided, or when the subsystems do not want to share detailed data of their inner state. In a distributed framework, a subsystem is aware of the connected subsystems, but has a very limited access to their state, hence avoiding both heavy communications and the sharing of sensible data between subsystems.

In this paper, we treat the coupling constraint (1b) using a *Lagrangian relaxation* [6] and decompose (1) into low-dimensional local subproblems that can be solved independently. The resulting dual function is once continuously differentiable but only piecewise twice differentiable. Lagrangian relaxtion is used in many different contexts to tackle convex large scale problems, e.g. the authors of [7] propose a coordinate ascent approach to solve constrained matrix problems. In [8], [9], [10], a gradient method, whereas in [11], [12], [13], a fast gradient method is used in order to attain dual optimality. All these methods make use of only first order derivatives to obtain a search direction and thus their theoretical and practical convergence cannot be faster than sublinear. The authors of [14], [15], [16] overcome this limitation in practise by using second-order derivatives in the dual space.

In [16], a distributed method based on dual decomposition is discussed for optimal control problems, where a non-smooth Newton step is used to perform the updates of the dual variables, and a Conjugate Gradient (CG) method is used to compute the dual step in a distributed fashion. In

[1] Attila Kozma and Moritz Diehl are with the Department of Electrical Engineering (ESAT), KU Leuven, 3001 Heverlee, Belgium {attila.kozma, moritz.diehl}@esat.kuleuven.be

[2] Emil Klintberg and Sebastien Gros are with the Department of Signals and Systems, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden {grosse,kemil}@chalmers.se

978-1-4799-3274-0/$31.00 ©2014 AACC

this approach, it has been observed that the presence of local inequality constraints, in particular state constraints, can yield singular dual Hessians during the dual iterations. A singular dual Hessian causes the CG method to fail, which then temporarily collapses to a classical gradient step. It has been observed that many gradient steps are needed before a full rank dual Hessian is recovered, which hinders seriously the convergence of the method. In this paper, we aim at tackling this issue by relaxing the local inequalities during the early phase of the dual iterations.

The paper is organised as follows. Section II presents the proposed dual decomposition with second-order information and the L_2 constraint relaxation. It presents a formal proof showing that the relaxation prevents the singularity of the dual Hessian. Section III presents the Non-Smooth Newton approach used in this paper which allows for performing the iterations on the dual variables in an almost completely distributed fashion. Section IV presents an example of application, where we experimentally show that the L_2 penalty can speed the convergence up.

Contribution: this paper presents a constraint relaxation strategy to prevent the problem of dual Hessian singularity in the second-order dual decomposition approach. The benefits of the method are formally established and numerically demonstrated.

II. Dual Decomposition with Second-order Information

In this section, the dual decomposition method is introduced together with a Newton strategy in the dual space.

A. Dual decomposition

We introduce the dual variables $\lambda_{i,j} \in \mathbb{R}^{v_{i,j}}$ for all $(i,j) \in E$ corresponding to constraint (1b). We define the Lagrange function as

$$\mathcal{L}(x,\lambda) := \sum_{k=1}^{N} \left(\frac{1}{2} x_k^T H_k x_k + c_k^T x_k \right) + \sum_{(i,j)\in\mathcal{G}} \lambda_{i,j}^T \left(A_{i,j} x_i - B_{i,j} x_j \right). \quad (2)$$

Here, $\lambda \in \mathbb{R}^v$, $\lambda^T = [\lambda_{i_1,j_1}^T, \ldots, \lambda_{i_Q,j_Q}^T]$, and $(i_1,j_1),\ldots,(i_Q,j_Q)$ is the enumeration of the edges in $\mathcal{G}$ following a fixed order.

Furthermore, let us denote the set of edges including vertex i by $\mathcal{N}_i,\ i \in E$, i.e.,

$$\mathcal{N}_i := \{k | (i,k) \in E \text{ or } (k,i) \in E\}, \quad (3)$$

and let $\lambda_{\mathcal{N}_i},\ i \in E$ denote the dual variables that affect subproblem i, i.e.,

$$\lambda_{\mathcal{N}_i} := \mathrm{vec}(\{\lambda_{k,j} | (k,j) \in \mathcal{N}_i\}). \quad (4)$$

Note that $\mathcal{L}(x,\lambda)$ is separable in x, i.e.

$$\mathcal{L}(x,\lambda) = \sum_{i=1}^{N} \mathcal{L}_i(x_i, \lambda_{\mathcal{N}_i}), \quad (5)$$

with

$$\mathcal{L}_i(x_i,\lambda_{\mathcal{N}_i}) := \frac{1}{2} x_i^T H_i x_i + c_i^T x_i + \sum_{(i,j)\in E} \lambda_{i,j}^T A_{i,j} x_i - \sum_{(j,i)\in E} \lambda_{j,i}^T B_{j,i} x_i. \quad (6)$$

Thus, one can evaluate the dual function $d(\lambda) := -\min_{x\in\mathcal{X}} \mathcal{L}(x,\lambda)$ in parallel as

$$d(\lambda) = -\sum_{i=1}^{N} \min_{x_i \in \mathcal{X}_i} \mathcal{L}_i(x_i, \lambda_{\mathcal{N}_i}). \quad (7)$$

Note that since (1) is strictly convex, $d(\lambda)$ is convex and continuously differentiable, but not twice differentiable. However, the second-order derivative of $d(\lambda)$, or dual Hessian, exists piecewise [17]. It has been observed in this context that the dual Hessian can be singular, even if the QP (1) is strictly convex and satisfies the Linear Independence Constraints Qualification (LICQ). This issue is studied and addressed next.

B. Singularity of the Dual Hessian

In this subsection, we discuss the occurrence of a singular dual Hessian in the framework described in this paper, and propose a remedy to address that issue.

We describe the conditions leading to the singularity of the dual Hessian. To that end, we write the dual problem in the compact form

$$d(\lambda) = -\begin{pmatrix} \min_x & \frac{1}{2} x^T H x + c^T x + \lambda^T C x \\ \text{s.t.} & Dx - e \le 0 \end{pmatrix}, \quad (8)$$

where C summarises all the coupling constraints in (2), and H, c, D, e gather the local H_k, c_k, D_k, e_k. The dual Hessian is then given by

$$\nabla_\lambda^2 d(\lambda) = -C \frac{\partial x}{\partial \lambda}, \quad (9)$$

where $\frac{\partial x}{\partial \lambda}$ is the solution of the linear system

$$\begin{bmatrix} H & D_{\mathcal{A}}^T \\ D_{\mathcal{A}} & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial \lambda} \\ \frac{\partial \mu_{\mathcal{A}}}{\partial \lambda} \end{bmatrix} = -\begin{bmatrix} C^T \\ 0 \end{bmatrix}. \quad (10)$$

Here, $\mathcal{A}$ reports the active set of inequality constraints, i.e. $D_{\mathcal{A}} x - e_{\mathcal{A}} = 0$. Let $N_{\mathcal{A}}$ be the null space of the active constraints, i.e. $D_{\mathcal{A}} N_{\mathcal{A}} = 0$. Then the solution $\frac{\partial x}{\partial \lambda}$ to (10) is given by

$$\frac{\partial x}{\partial \lambda} = -N_{\mathcal{A}} \left(N_{\mathcal{A}}^T H N_{\mathcal{A}} \right)^{-1} N_{\mathcal{A}}^T C^T, \quad (11)$$

and the dual Hessian reads as

$$\nabla_\lambda^2 d(\lambda) = C N_{\mathcal{A}} \left(N_{\mathcal{A}}^T H N_{\mathcal{A}} \right)^{-1} N_{\mathcal{A}}^T C^T. \quad (12)$$

It follows that the dual Hessian becomes singular if $CN_{\mathcal{A}}$ is rank deficient. In the context of the original problem (1), this situation occurs if the coupling constraints (1b) together with the active local inequality constraints (1c) are not linearly independent. Thus, certain values of λ can yield a singular dual Hessian, which forces the algorithm proposed in [16] to make gradient steps.

In [16], we have used $d(\lambda)$ directly, which may result in numerous gradient steps, instead of the more efficient Newton steps. In order to avoid this phenomenon, we introduce a *modified dual function*, which uses penalties, instead of the hard local inequality constraints. More precisely, we regard

$$d_\gamma(\lambda) := -\sum_{i=1}^{N} \begin{pmatrix} \min_{x_i} & \mathcal{L}_i(x_i, \lambda_{\mathcal{N}_i}) + \frac{\gamma}{2}\|s_i\|_2^2 \\ \text{s.t.} & D_i x_i - e_i \leq s_i \end{pmatrix}. \quad (13)$$

Here, $\gamma \in \mathbb{R}$ is the weight of the L_2 penalty of the constraints, and $s_i \in \mathbb{R}^{w_i}$ is a slack variable. Note that once $\gamma = 0$, the local inequalities are completely relaxed and thus not effective, whereas if $\gamma \to \infty$ the local inequalities are becoming more and more important, i.e. less and less relaxed. Moreover, in the limit, the modified dual $d_\infty(\lambda)$ is equivalent to $d(\lambda)$. Observe that the positivity of the slack variables does not need to be enforced. Indeed, it can be easily verified that inequality constraints of the form $s_i \geq 0$ included in (13) would never be strictly active, and can therefore be discarded.

The L_2 relaxation proposed in (13) offers a remedy to render the dual Hessian non-singular. The following lemma establishes the effect of the relaxation.

Lemma 1: Assume that C is fullrank, for any λ and $\gamma > 0$, the Hessian of the modified dual function $d_\gamma(\lambda)$ is non-singular.

Proof: We write the modified dual problem in the compact form

$$d_\gamma(\lambda) = -\begin{pmatrix} \min_x & \frac{1}{2}x^T H x + c^T x + \frac{\gamma}{2} s^T s + \lambda^T C x \\ \text{s.t.} & Dx - e \leq s \end{pmatrix}.$$

The dual Hessian reads

$$\nabla_\lambda^2 d_\gamma(\lambda) = -C\frac{\partial x}{\partial \lambda}, \quad (14)$$

where the sensitivity $\frac{\partial x}{\partial \lambda}$ is the solution of the linear system

$$\begin{bmatrix} H & D_{\mathcal{B}}{}^T & 0 \\ D_{\mathcal{B}} & 0 & -I_{\mathcal{B}} \\ 0 & -I_{\mathcal{B}}{}^T & \gamma I \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial \lambda} \\ \frac{\partial \mu_{\mathcal{B}}}{\partial \lambda} \\ \frac{\partial s}{\partial \lambda} \end{bmatrix} = -\begin{bmatrix} C^T \\ 0 \\ 0 \end{bmatrix}. \quad (15)$$

Here, $I_{\mathcal{B}}$ is a full row rank matrix with single unitary entries at the row indices reported by $\mathcal{B}$. We use the fact that $I_{\mathcal{B}} I_{\mathcal{B}}^T = I$ to eliminate the last row in (15) and obtain

$$\begin{bmatrix} H & D_{\mathcal{B}}{}^T \\ D_{\mathcal{B}} & -\gamma^{-1} I \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial \lambda} \\ \frac{\partial \mu_{\mathcal{B}}}{\partial \lambda} \end{bmatrix} = -\begin{bmatrix} C^T \\ 0 \end{bmatrix}, \quad (16)$$

which has the solution

$$\frac{\partial x}{\partial \lambda} = -\left(H + \gamma D_{\mathcal{B}}{}^T D_{\mathcal{B}}\ \right)^{-1} C^T. \quad (17)$$

The dual Hessian therefore reads as

$$\nabla_\lambda^2 d_\gamma(\lambda) = C\left(H + \gamma D_{\mathcal{B}}{}^T D_{\mathcal{B}}\ \right)^{-1} C^T, \quad (18)$$

which is non-singular since C is full row rank. ■

III. Non-smooth Newton Method in the Dual Space

In this section, we revise the main algorithm presented in [16] when applied to $d_\gamma(\lambda)$. The optimal dual variables for a fixed γ can be found as the solution of the unconstrained dual optimization problem defined by

$$\min_\lambda d_\gamma(\lambda). \quad (19)$$

Since this is a convex, non-smooth, once continuously differentiable problem, we utilize a *non-smooth Newton method* [18], [19], while driving γ to a sufficiently large number. In each iteration of the non-smooth Newton method, a second-order model of $d_\gamma(\lambda)$ in $\lambda^{(k)}$ is minimized that is

$$m(p) = d_\gamma(\lambda^{(k)}) + \nabla d_\gamma(\lambda^{(k)})^T p + \frac{1}{2}p^T \hat{H}^{(k)} p, \quad (20)$$

where $\hat{H}_\gamma^{(k)} \in \partial\left(\nabla d_\gamma(\lambda^{(k)})\right)$ in the sense of Clarke [20]. Solving $p^{(k)} := \arg\min_p m(p)$ yields a descent direction in the space of λ and can be obtained as the solution of the linear system

$$\hat{H}_\gamma^{(k)} p + \nabla d_\gamma(\lambda^{(k)}) = 0. \quad (21)$$

This Newton system, in the view of Lemma 1, is non-singular and can be solved inexactly by a conjugate gradient (CG) method [21].

Algorithm 1: Singularity-free Dual Newton-CG method

```
Input : $\lambda^{(0)}$, $\gamma > 0$, $\tau > 1$, $\epsilon_1 > 0$, $\epsilon_2 > 0$
while $\|\nabla d_\gamma(\lambda)\| > \epsilon_1$ or $\|s\| > \epsilon_2$ do
    Compute $\nabla d_\gamma(\lambda)$ and $\nabla^2 d_\gamma(\lambda)$ in a distributed fashion.
    Solve $\hat{H}^{(k)} p + \nabla d_\gamma(\lambda^{(k)}) = 0$ with CG.
    Find proper stepsize $t$.
    $\lambda^{(k+1)} := \lambda^{(k)} + tp$
    $\gamma := \tau \cdot \gamma$
    $k := k + 1$
end
return $\lambda^{(k)}$
```

In Algorithm 1, we have summarized the most important steps of the proposed approach. Observe that termination takes place if a dual optimal and a primal feasible solution is found up to accuracy ϵ_1 and ϵ_2, respectively. In the following, we discuss each step in more detail.

A. Calculation of the dual sensitivities

In the proposed approach, the knowledge of the dual derivatives are necessary, thus we show that both the gradient and the Hessian is computable via local exchange of contributions.

The dual function $d_\gamma(\lambda)$ is continuously differentiable [22, p. 100] and its derivative in the direction of $\lambda_{i,j}$ is given by

$$\frac{\partial d_\gamma(\lambda)}{\partial \lambda_{i,j}}^T = -A_{i,j} x_i^*(\lambda_{\mathcal{N}_i}) + B_{i,j} x_j^*(\lambda_{\mathcal{N}_j}), \quad (22)$$

where $x_i^*(\lambda_{\mathcal{N}_i}) := \arg\min_{x_i\in\mathcal{X}_i} \mathcal{L}_i(x_i, \lambda_{\mathcal{N}_i}) + \frac{\gamma}{2}s_i^2$. It should be understood that only subproblems i and j need to be solved in order to calculate the dual gradient in the direction of $\lambda_{i,j}$. Moreover, this slice of the gradient can be computed in subproblems i and j by a simple local exchange of contributions. This is one of the reasons why first order methods are often used in a dual decomposition framework.

We obtain an $\hat{H}^{(k)}$ by differentiating (22). It is important to observe that the dual Hessian has a well-defined sparsity structure, which is essentially determined by the interconnections of subsystems. The block corresponding to variables $\lambda_{i,j},\ (i,j)\in E$ can be written as

$$\frac{\partial^2 d(\lambda)}{\partial\lambda_{i,j}\partial\lambda_{i,j}} = -A_{i,j}\frac{\partial x_i^*(\lambda_{\mathcal{N}_i})}{\partial\lambda_{i,j}} + B_{i,j}\frac{\partial x_j^*(\lambda_{\mathcal{N}_j})}{\partial\lambda_{i,j}}. \tag{23a}$$

The off-diagonal blocks corresponding to the row of $\lambda_{i,j}$ and the column of $\lambda_{k,l},\ (i,j)\neq(k,l)$ are given by

$$\frac{\partial^2 d(\lambda)}{\partial\lambda_{i,j}\partial\lambda_{k,l}} = \begin{cases} -A_{i,j}\frac{\partial x_i^*(\lambda_{\mathcal{N}_i})}{\partial\lambda_{k,l}} & \text{if}(k,l)\in\mathcal{N}_i\setminus\mathcal{N}_j, \\ B_{i,j}\frac{\partial x_j^*(\lambda_{\mathcal{N}_j})}{\partial\lambda_{k,l}} & \text{if}(k,l)\in\mathcal{N}_j\setminus\mathcal{N}_i, \\ 0 & \text{otherwise}. \end{cases} \tag{23b}$$

In other words, the second derivatives with respect to $\lambda_{i,j}$ and $\lambda_{k,l}$ are nonzero if and only if edge (k,l) is connected to the node i or j.

Now, let us compute the sensitivity of the solution in subproblem i with respect to all influential dual variables, i.e. $\frac{\partial x_i^*(\lambda_{\mathcal{N}_i})}{\partial\lambda_{\mathcal{N}_i}}$. To this end, we follow the procedure presented in Lemma 1. Let $\mathcal{A}_i$ report the active set at the actual optimal solution for problem i. For notational convenience, we introduce matrix C_i such that

$$\lambda_{\mathcal{N}_i}C_i x_i := \sum_{(i,j)\in E}\lambda_{i,j}^T A_{i,j}x_i - \sum_{(j,i)\in E}\lambda_{j,i}^T B_{j,i}x_i. \tag{24}$$

The optimal primal-dual solution of subproblem i is the solution of

$$\begin{bmatrix} H_i & 0 & D_{i,\mathcal{A}_i}^T \\ 0 & \gamma & -I_{\mathcal{A}_i}^T \\ D_{i,\mathcal{A}_i} & -I_{\mathcal{A}_i} & 0 \end{bmatrix} \begin{bmatrix} x_i^*(\lambda_{\mathcal{N}_i}) \\ s_i^* \\ \mu_i^* \end{bmatrix} = -\begin{bmatrix} c_i + C_i^T\lambda_{\mathcal{N}_i} \\ 0 \\ -e_{i,\mathcal{A}_i} \end{bmatrix}, \tag{25}$$

where μ_i^* is the optimal dual variable corresponding to the constraints reported by $\mathcal{A}_i$. It can be easily verified that

$$\frac{\partial x_i^*(\lambda_{\mathcal{N}_i})}{\partial\lambda_{\mathcal{N}_i}} = \left(\gamma D_{i,\mathcal{A}_i}^T D_{i,\mathcal{A}_i} + H_i\right)^{-1} C_i^T. \tag{26}$$

Note that the cost of this operation is dominated by the matrix inversion, which is carried out locally in each subproblem. Also, the local contributions to (23) are directly computable.

B. Products between vectors and the dual Hessian

Since the Newton system is solved via a CG procedure, the product of the Hessian with a vector is of major interest. We consider first rows in the dual Hessian corresponding to $\lambda_{i,j}$ that is

$$\begin{aligned} \frac{\partial^2 d(\lambda)}{\partial\lambda_{i,j}\partial\lambda}p = \sum_{(k,l)\in\mathcal{N}_i} & -A_{i,j}\frac{\partial x_i^*(\lambda_{\mathcal{N}_i})}{\partial\lambda_{k,l}}p_{k,l} \\ & + \sum_{(k,l)\in\mathcal{N}_j} B_{i,j}\frac{\partial x_j^*(\lambda_{\mathcal{N}_j})}{\partial\lambda_{k,l}}p_{k,l}. \end{aligned} \tag{27}$$

Here, $p\in\mathbb{R}^v,\ p^T = [p_{i_1,j_1}^T,\dots,p_{i_Q,j_Q}^T]$ and the order $(i_1,j_1),\dots,(i_Q,j_Q)$ corresponds to the one defined in λ. It should be noted that the vector slice $\frac{\partial^2 d(\lambda)}{\partial\lambda_{i,j}\partial\lambda}p$ can be calculated on both nodes i and j via only local exchange of contributions to the sum in (27). In fact, in each CG iteration, a local exchange of vectors, i.e. a matrix-vector product, takes place in between subproblem i and j for each $(i,j)\in E$.

We use (27) to calculate the quadratic form $p^T\frac{\partial^2 d(\lambda)}{\partial\lambda^2}p$, i.e.

$$\begin{aligned} p^T\frac{\partial^2 d(\lambda)}{\partial\lambda^2}p &= \sum_{(i,j)\in E} p_{i,j}^T\frac{\partial^2 d(\lambda)}{\partial\lambda_{i,j}\partial\lambda}p = \\ &= \sum_{(i,j)\in E} p_{i,j}^T\left[\sum_{(k,l)\in\mathcal{N}_i} -A_{i,j}\frac{\partial x_i^*(\lambda_{\mathcal{N}_i})}{\partial\lambda_{k,l}}p_{k,l}\right. \\ &\left.+ \sum_{(k,l)\in\mathcal{N}_j} B_{i,j}\frac{\partial x_j^*(\lambda_{\mathcal{N}_j})}{\partial\lambda_{k,l}}p_{k,l}\right]. \end{aligned} \tag{28}$$

Note that in (28) a summation over all $(i,j)\in E$ takes place. Each term of this sum can be calculated via local communication as mentioned earlier. However, the calculation of the sum itself requires a global operation, such that all local contributions are collected and the result is broadcasted. This operation is often referred to as a *global summation* [23].

Based upon the previous discussion the residual $\hat{H}p + \nabla d(\lambda)$ is computable in slices with respect to $\lambda_{i,j}$ and is available in subproblems i and j.

C. Conjugate gradient method and line-search strategy

For algorithmic details on how to solve the Newton system (21), we refer to Algorithm 2 in [16].

Upon obtaining a search direction $p^{(k)}$, i.e., an approximate solution of (21), one can perform the following update of the dual variables

$$\lambda^{(k+1)} := \lambda^{(k)} + t^{(k)}p^{(k)}, \tag{29}$$

where $t^{(k)}$ is an appropriately chosen stepsize. To this end, we use an Armijo line search strategy with backtracking. In each iteration of the Newton method, we would like to find a $t^{(k)}$ such that

$$d_\gamma(\lambda^{(k)} + t^{(k)}p^{(k)}) \leq d_\gamma(\lambda^{(k)}) + \sigma t^{(k)}\nabla d_\gamma(\lambda^{(k)})^T p^{(k)}, \tag{30}$$

holds. For further algorithmic details we refer to Algorithm 1 in [16].

IV. Numerical Example: Optimal Control of Connected Masses

The numerical experiments are carried out on an academic optimal control problem (OCP). We consider a linear model of a unidimensional chain consisting of connected masses. Each mass $n \in \{1, \dots, N\}$ is described by its position $p_n \in \mathbb{R}$ and its velocity $v_n \in \mathbb{R}$ and can be controlled via a force F_n. For the sake of simplicity, we directly present the equations obtained by using the distributed multiple shooting method, i.e., after discretization. To that end, a single step of the explicit Euler method is used. Let Δt and M denote the length of time intervals and the number of time intervals, respectively. The system dynamics are enforced by the constraints

$$\begin{bmatrix} p_n^{(m+1)} \\ v_n^{(m+1)} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 \\ -2\Delta t & 1 & \Delta t & \Delta t & \Delta t \end{bmatrix} \begin{bmatrix} p_n^{(m)} \\ v_n^{(m)} \\ z_{n,n-1}^{(m)} \\ z_{n,n+1}^{(m)} \\ F_n^{(m)} \end{bmatrix}. \tag{31a}$$

Here, $m = 1, \dots, M-1$, $n = 1, \dots, N$ and the variables $z_{n,n-1}^{(m)}$ and $z_{n,n+1}^{(m)}$ belong to the masspoint n on time interval m and represent a finite discretization, e.g. polynomial coefficients, of the positions of masspoints $n-1$ and $n+1$, respectively. The spatial coupling between the mass points is ensured by

$$z_{n,n+1}^{(m)} = p_{n+1}^{(m)} \quad n = 1, \dots, N-1, \tag{31b}$$

$$z_{n,n-1}^{(m)} = p_{n-1}^{(m)} \quad n = 2, \dots, N. \tag{31c}$$

The objective function penalizes the deviation from the steady state 0 using the L_2-norm. The bounds on the forces are given by

$$\underline{F} \le F_n^{(m)} \le \overline{F}, \quad m = 1, \dots, M, \ n = 1, \dots, N, \tag{32a}$$

whereas the positions and velocities are constrained by

$$\underline{p}_n^{(m)} \le p_n^{(m)} \le \overline{p}_n^{(m)}, \quad \underline{v}_n^{(m)} \le v_n^{(m)} \le \overline{p}_n^{(m)}, \tag{32b}$$
$$m = 2, \dots M, \quad n = 1, \dots, N.$$

Furthermore, the first and last mass points are fixed

$$p_1^m = p_{\text{first}}, \quad p_n^m = p_{\text{last}}, \quad m = 1, \dots, M. \tag{33}$$

Let us introduce $S := NM$ and for all $n = 1, \dots, N$, and $m = 1, \dots, M$, and let

$$y_{(m-1)N+n}^T := (p_n^{(m)}, v_n^{(m)}, z_{n,n-1}^{(m)}, z_{n,n+1}^{(m)}, F_n^{(m)})^T. \tag{34}$$

We summarize the discretized OCP as a convex QP of the form

$$\min_{y_1, \cdots y_S} \frac{1}{2} \sum_{k=1}^{S} y_k^T R_k y_k \tag{35a}$$

$$\text{s.t } C_{i,j} y_i = D_{i,j} y_j, \ (i,j) \in \overline{E} \tag{35b}$$

$$y_l \in \mathcal{Y}_l, \ l = 1, \dots, S. \tag{35c}$$

Here, $(i,j) \in \overline{E}$ if and only if $j = i + N$ or ($j = i + 1$ and $j \mod N \neq 0$). Moreover, $C_{i,j}$ and $D_{i,j}$ can be directly computed from (31) and $\mathcal{Y}_l$ can be derived from (32).

TABLE I

Solution statistics with different initial guesses to achieve convergence.

Init. guess	Newt. it.	QP sol.	CG it.	RFG
random	46	92	1261	14045
$\mathbf{1}$	46	92	1171	9877
$2 \cdot \mathbf{1}$	46	92	1183	10151
$3 \cdot \mathbf{1}$	46	92	1189	11619

Note that (35) has the same structure as (1) and thus the proposed algorithm is applicable. We generated an instance of (35) with $m = 20$ mass points and $n = 15$ time intervals resulting in a decomposable QP with 320 subproblems. The full QP has a total of 1540 primal variables, 1170 dual variables, and 2340 inequality constraints.

The proposed stopping criterion was $\|\nabla d_\gamma(\lambda)\|_\infty < 10^{-5}$ and $\|s\|_\infty < 10^{-5}$. The convergence of our algorithm was tested with many different initial guesses, including random initialization. In each test case, convergence was attained in 46 Newton iterations taking only full steps. See Table I, where the number of Newton iterations, the number of local QP solutions per subproblem, and the overall number of CG iterations is reported for a couple of test cases. In the last column, one can observe the number of dual iterations made by a state-of-the-art restarted fast gradient method (RFG) [24], applied in the dual space. We note that the overall computational cost of one dual iteration with our proposed method is somewhat more than one of the RFG method due to more communication and local linear algebraic operations. It was also observed that in all test cases, the original method without the relaxation technique was never successful and always got stuck due to a singular dual Hessian.

We show one test case, where the initial guess was chosen randomly, and plotted the evolution of $\|d_\gamma(\lambda)\|$ and $\|s\|$ with the proposed method versus $\|d(\lambda)\|$ given by the approach with hard constraints in Figure 1.

We conclude that for this particular example, the number of iterations and thus the number of local QP solutions is an order of magnitude smaller than with a first order method. Moreover, the use of L_2 penalty on the constraints accelerates the practical convergence speed of the dual Newton-CG approach.

V. Conclusions and Future Work

In this paper, we consider the problem of solving distributed Quadratic Programs efficiently, and improve on the method proposed in [16] where a dual decomposition approach alongside a distributed Newton iteration based on the Conjugate Gradient method is used, yielding faster convergence than first-order techniques. An issue has been observed in the original method resulting in a rank deficiency of the dual Hessian, which forces the Newton iteration to

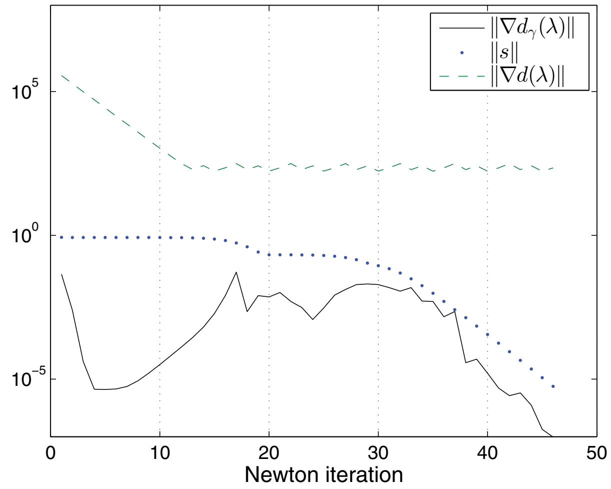

Fig. 1. Illustration of the relaxation-based approach (plain line) versus the original method (dashed line), initialized at a random point. The dotted line displays the convergence of the slack variables, resulting from gradually increasing the γ parameter over the dual iterations.

collapse into a simple gradient step for many dual iterations. This paper investigates that issue, explains it, and proposes an L_2 relaxation of the local inequality constraints to address it. A formal proof is proposed, showing that the relaxation prevents the rank deficiency problem.

Future work will consider improving the constraints relaxation using a barrier approach. The possibility of designing a heuristic to tune the barrier parameters efficiently as the dual iteration converges will be investigated. Early numerical results suggest that the relaxation strategy improves significantly the convergence rate of the second-order dual decomposition algorithm. This effect will be further studied.

ACKNOWLEDGEMENTS

This research was supported by Research Council KUL: PFV/10/002 Optimization in Engineering Center OPTEC, GOA/10/09 MaNet and GOA/10/11 Global real- time optimal control of autonomous robots and mechatronic systems. Flemish Government: IOF/KP/SCORES4CHEM, FWO: PhD/postdoc grants and projects: G.0320.08 (convex MPC), G.0377.09 (Mechatronics MPC); IWT: PhD Grants, projects: SBO LeCoPro; Belgian Federal Science Policy Office: IUAP P7 (DYSCO, Dynamical systems, control and optimization, 2012-2017); EU: FP7-EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC ST HIGHWIND (259 166), Eurostars SMART, ACCM.

REFERENCES

[1] M. Powell, "Algorithms for nonlinear constraints that use Lagrangian functions," *Mathematical Programming*, vol. 14, no. 3, pp. 224–248, 1978.

[2] H. Bock and K. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," in *Proceedings 9th IFAC World Congress Budapest*. Pergamon Press, 1984, pp. 242–247.

[3] C. Savorgnan, C. Romani, A. Kozma, and M. Diehl, "Multiple shooting for distributed systems with applications in hydro electricity production," *Journal of Process Control*, vol. 21, pp. 738–745, 2011.

[4] E. Gertz and S. Wright, "Object-Oriented Software for Quadratic Programming," *ACM Transactions on Mathematical Software*, vol. 29, no. 1, pp. 58–81, 2003.

[5] IBM Corp., *IBM ILOG CPLEX V12.1, User's Manual for CPLEX*, 2009.

[6] C. Lemaréchal, "Lagrangian relaxation," in *Computational Combinatorial Optimization*, ser. Lecture Notes in Computer Science, M. Jünger and D. Naddef, Eds. Springer Berlin / Heidelberg, 2001, vol. 2241, pp. 112–156. [Online]. Available: http://dx.doi.org/10.1007/3-540-45586-8_4

[7] R. W. Cottle, S. G. Duvall, and K. Zikan, "A lagrangean relaxation algorithm for the constrained matrix problem," *Naval Research Logistics Quarterly*, vol. 33, no. 1, pp. 55–76, 1986. [Online]. Available: http://dx.doi.org/10.1002/nav.3800330106

[8] L. S. Lasdon, *Optimization theory for Large Systems*. Dover, 1970.

[9] P. Tseng, "Dual ascent methods for problems with strictly convex costs and linear constraints: A unified approach," *SIAM Journal on Control and Optimization*, vol. 28, no. 1, pp. 214–242, 1990. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/0328011

[10] P. Giselsson and A. Rantzer, "Distributed model predictive control with suboptimality and stability guarantees," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, 2010, pp. 7272–7277.

[11] S. Richter, M. Morari, and C. Jones, "Towards computational complexity certification for constrained MPC based on Lagrange Relaxation and the fast gradient method," in *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011, pp. 5223 –5229.

[12] I. Necoara, V. Nedelcu, and I. Dumitrache, "Parallel and distributed optimization methods for estimation and control in networks," *Journal of Process Control*, vol. 21, no. 5, pp. 756 – 766, 2011, special Issue on Hierarchical and Distributed Model Predictive Control. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S095915241000257X

[13] P. Giselsson, M. D. Doan, T. Keviczky, B. D. Schutter, and A. Rantzer, "Accelerated gradient methods and dual decomposition in distributed model predictive control," *Automatica*, vol. 49, no. 3, pp. 829 – 833, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109813000101

[14] H. Ferreau, A. Kozma, and M. Diehl, "A parallel active-set strategy to solve sparse parametric quadratic programs arising in MPC," in *Proceedings of the 4th IFAC Nonlinear Model Predictive Control Conference, Noordwijkerhout, The Netherlands*, 2012.

[15] J. V. Frasch, M. Vukov, H. Ferreau, and M. Diehl. (2013) A dual Newton strategy for the efficient solution of sparse quadratic programs arising in SQP-based nonlinear MPC. Optimization Online 3972. [Online]. Available: http://www.optimization-online.org/DB_HTML/2013/07/3972.html

[16] A. Kozma, J. V. Frasch, and M. Diehl, "A Distributed Method for Convex Quadratic Programming Problems Arising in Optimal Control of Distributed Systems," in *Proceedings of the 52nd Conference on Decision and Control (CDC)*, 2013.

[17] W. Li and J. Swetits, "A new algorithm for solving strictly convex quadratic programs," *SIAM Journal of Optimization*, vol. 7, no. 3, pp. 595–619, 1997.

[18] L. Qi and J. Sun, "A nonsmooth version of Newton's method," *Mathematical Programming*, vol. 58, pp. 353–367, 1993.

[19] M. Gerdts and M. Kunkel, "A nonsmooth Newton's method for discretized optimal control problems with state and control constraints," *Journal of Industrial and Management Optimization*, vol. 4, pp. 247–270, 2008.

[20] F. Clarke, *Optimization and Nonsmooth Analysis*, ser. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1990.

[21] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Carnegie Mellon University, Pittsburgh, PA, USA, Tech. Rep., 1994.

[22] D. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: Numerical methods*. Prentice Hall, 1989.

[23] J. Lewis and R. van de Geijn, "Distributed memory matrix-vector multiplication and conjugate gradient algorithms," in *Supercomputing '93. Proceedings*, nov. 1993, pp. 484 – 492.

[24] B. O´Donoghue and E. Candès, "Adaptive restart for accelerated gradient schemes," *Foundations of Computational Mathematics*, pp. 1–18, 2013.