Noname manuscript No.
(will be inserted by the editor)

A dual Newton strategy with fixed iteration complexity for multi-stage MPC

E. Klintberg · D. Kouzoupis · M. Diehl · S. Gros

Received: date / Accepted: date

Abstract This paper considers the problem of solving Quadratic Programs (QPs) in the context of multi-stage Model Predictive Control (MPC). A Newton strategy is considered on the dual of the problem to achieve a parallelizable method. In this context, it has been observed that the globalization strategy can be expensive. In this paper, we propose to dualize both the non-anticipativity constraints and the dynamics in order to obtain a computationally cheap globalization. The dual Newton system is then reformulated to small highly structured linear systems that to a large extent can be solved in parallel.

Keywords Robust control · Multi-stage MPC · Dual Newton strategy

1 Introduction

Model Predictive Control (MPC) is the preferred control technique in a growing set of applications due to the natural way in which constraints can be incorporated in the control policy. However, constraint satisfaction can in general not be guaranteed if uncertainties are present in the system. Several methods have been proposed to handle uncertainties in the employed model, e.g. min-max MPC [6] and tube-based MPC [21], [20]. Another common formulation, often labeled multi-stage MPC or stochastic programming, represents the uncertainty via a finite number of realizations at each decision point [23], [1], [16].

E. Klintberg
E-mail: kemil@chalmers.se / emil.klintberg@qamcom.se

D. Kouzoupis
E-mail: dimitris.kouzoupis@imtek.uni-freiburg.de

M. Diehl
E-mail: moritz.diehl@imtek.uni-freiburg.de

S. Gros
E-mail: grosse@chalmers.se

This formulation posses attractive properties such as recursive feasibility [18], and increased feasibility [17]. A major drawback, however, is the size of the underlying optimization problem, which grows exponentially with the length of the MPC control horizon.

Computational aspects of robust MPC have been considered e.g. in [9], where a primal-dual interior point method is proposed for min-max MPC, and in [26], where real-time feasibility is achieved via a tube-based robust MPC formulation together with the early termination of an interior point method. Moreover, several optimization methods have been proposed in the context of multi-stage MPC. The authors of e.g. [4], [11], [24] propose to use tailored interior point methods, the authors of [15], [12] use parallelizable active-set methods, whereas the authors of [3], [22], [19] use various decomposition techniques in order to exploit the intrinsic structure of the problem.

The active-set methods in [15], [12] are particularly well suited for multi-stage MPC, due to the natural way in which the similarity between subsequent Quadratic Programs (QPs) can be exploited, More specifically, in the case the active set does not change between subsequent instances of the multi-stage MPC problem, the methods converge in one step. However, the methods suffer from two drawbacks stemming from the non-smooth dual problem. First, there is no practically useful upper bound on the number of iterations needed to solve a problem. This is a limitation in safety critical applications, although methods of this kind usually work well in practice [7], [8]. Secondly, globalization can be expensive since many back-tracking steps may be required at each iteration to enforce convergence. This can be a significant drawback since every step involves solving QPs which become the computational bottleneck of the method. In this paper, we overcome the second difficulty for the practically important class of multi-stage MPC problems with a diagonal cost and simple bounds.

In contrast to the methods proposed in [15], [12], we dualize both the non-anticipativity constraints and the dynamical constraints. As a result, the solution of the the QPs becomes computationally negligible, whereas the resulting dual Newton system becomes larger. To enhance the solution of the Newton system it is reformulated into several small highly structured linear systems, which to a large extent can be solved in parallel. Consequently, computational effort is moved from the solution of the QPs to the solution of the Newton system. This is an improvement since the Newton system is only solved once every iteration whereas the QPs may be solved multiple times. As a result, we obtain a dual Newton strategy with a computational cost per iteration that is almost constant. For cases where a considerable amount of back-tracking is performed, our method saves a significant amount of computations.

The paper is organized as follows. In Section 2, we recall multi-stage MPC and the non-smooth dual Newton strategy. In Section 3, we detail the parallel calculation of the dual Hessian and the dual gradient, whereas in Section 4, we propose a reformulation of the Newton system to exploit the problem specific structure and enhance parallel calculations. The resulting method is presented

in Section 5. In Section 6, a numerical experiment is presented. The paper is concluded in Section 7.

2 Preliminaries

In this section, we recall multi-stage MPC and Newton strategies in the context of dual decomposition.

2.1 Multi-stage MPC

As a result of imperfect models and uncertain disturbances, constraint satisfaction can in general not be guaranteed for MPC schemes. A common remedy, often denoted as multi-stage MPC or stochastic programming, is to discretize the underlying stochastic process, and describe the evolution of the uncertainty via a scenario tree [23]. To that end, we consider a discrete-time, constrained system with uncertain parameters θ:

$$x_{i+1} = A(\theta)x_i + B(\theta)u_i \tag{1a}$$
$$x_{\min} \leq x_i \leq x_{\max} \tag{1b}$$
$$u_{\min} \leq u_i \leq u_{\max} \tag{1c}$$

where $x_i \in \mathbb{R}^n$ and $u_i \in \mathbb{R}^m$ denote the state and control variables respectively. To account for the uncertain parameters, we consider m_d realizations of (1) at each time stage. The evolution of the system can then be described by a scenario-tree as depicted in Figure 1.

We define a scenario as a path from the root node to a leaf node of the scenario tree. The number of scenarios is thus growing exponentially with the length of the MPC horizon, yielding very large optimization problems. It is therefore often proposed to treat the uncertain parameters as constant after a certain period of time. We denote the time period where the parameters can change as the robust horizon N_r, in contrast to the prediction horizon N. Accordingly, we consider $M = m_d^{N_r}$ scenarios.

To enhance parallel computations, we introduce separate state and control variables for each scenario, i.e. we introduce $x_k = [x_{k,1}^T \cdots x_{k,N}^T]^T \in \mathbb{R}^{\bar{n}}$, with $x_{k,i} \in \mathbb{R}^n$, and $u_k = [u_{k,0}^T \cdots u_{k,N-1}^T]^T \in \mathbb{R}^{\bar{m}}$, with $u_{k,i} \in \mathbb{R}^m$ for $k = 1, \ldots, M$. However, because the uncertainty cannot be anticipated, control actions are restricted to only depend on historical realizations of the uncertainty, such that the control variables of the scenarios are coupled at their shared nodes. More specifically, if the uncertainty realizations for scenario k and l are identical up to and including time stage i, their control inputs should be identical up to that time stage, i.e. $u_{k,j} = u_{l,j},\ \forall j = 0, \cdots i$. This restriction is commonly denoted as *non-anticipativity constraints*. The resulting MPC problem can be

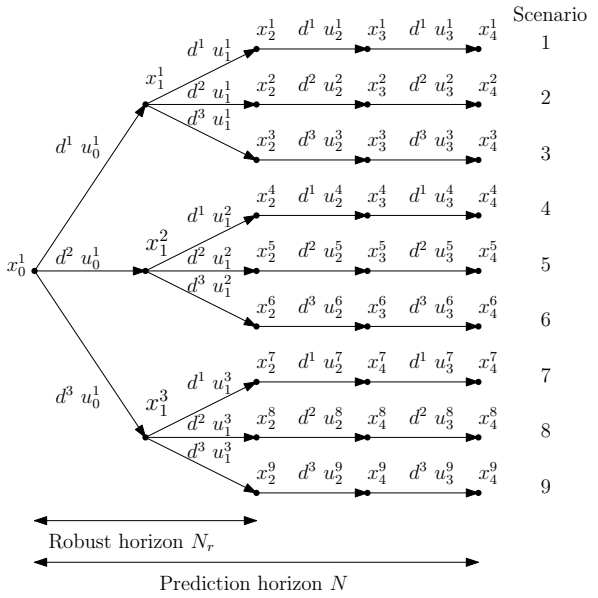

Fig. 1: The evolution of the system represented as a scenario tree. For nodes and branches that are shared between multiple scenarios, the variable corresponding to the scenario with the lowest index is visualized in the tree.

formulated as:

$$\min_{x,u} \quad \sum_{k=1}^{M} V_k(x_k, u_k) \tag{2a}$$

$$\text{s.t.} \quad \sum_{k=1}^{M} \bar{C}_k u_k = 0 \tag{2b}$$

$$\bar{A}_k x_k + \bar{B}_k u_k = b_k, \quad k = 1, \dots, M \tag{2c}$$

$$\underline{x}_k \leq x_k \leq \overline{x}_k, \quad k = 1, \dots, M \tag{2d}$$

$$\underline{u}_k \leq u_k \leq \overline{u}_k, \quad k = 1, \dots, M \tag{2e}$$

where we have introduced the notations $x = [x_1^T \dots x_M^T]^T$ and $u = [u_1^T \dots u_M^T]^T$ for the collection of variables over the scenarios. Additionally, we have defined $V_k(x_k, u_k) = \frac{1}{2} x_k^T \bar{Q}_k x_k + \frac{1}{2} u_k^T \bar{R}_k u_k + \bar{q}_k^T x_k + \bar{r}_k^T u_k$, $b_k = [-\bar{x}^T A_k^T \ 0 \dots 0]^T$,

where $\bar{x}$ denotes the initial state estimate, and:

$$\bar{Q}_k = \begin{bmatrix} Q_{k,1} & & \\ & \ddots & \\ & & Q_{k,N} \end{bmatrix}, \; \bar{R}_k = \begin{bmatrix} R_{k,0} & & \\ & \ddots & \\ & & R_{k,N-1} \end{bmatrix} \tag{3a}$$

$$\bar{A}_k = \begin{bmatrix} -I & & & \\ A_{k,1} & -I & & \\ & \ddots & \ddots & \\ & & A_{k,N-1} & -I \end{bmatrix}, \; \bar{B}_k = \begin{bmatrix} B_{k,0} & & \\ & \ddots & \\ & & B_{k,N} \end{bmatrix} \tag{3b}$$

where we require the matrices $\bar{Q}_k \in \mathbb{S}^{nN}_{++}$ and $\bar{R}_k \in \mathbb{S}^{mN}_{++}$ to be diagonal.

The formulation of the non-anticipativity constraints (2b) provides some freedom in constructing the sparsity structure of $\bar{C}$ via the ordering of the constraints. The structure of $\bar{C}$ has a direct impact on the sparsity structures of the dual Hessian, and ought to be exploited in order to facilitate the solving of (2). Due to the fact that the non-anticipativity constraints are enforced at nodes in the scenario tree, and that neighboring scenarios share nodes, we propose to enforce the non-anticipativity constraints in a chain structure. To that end we introduce the notation:

$$p = m \sum_{k=1}^{M-1} n_{c,(k,k+1)} \tag{4}$$

where $n_{c,(k,k+1)}$ denotes the number of common nodes in the scenario tree for scenario k and $k+1$, and select the matrices $\bar{C}_k \in \mathbb{R}^{p \times \bar{m}}$ as follows:

$$\begin{aligned} \bar{C} &= \left[\begin{array}{c|c|c|c|c} C_{1,2} & -C_{1,2} & & & \\ & C_{2,3} & -C_{2,3} & & \\ & & \ddots & \ddots & \\ & & & C_{M-1,M} & -C_{M-1,M} \end{array}\right] = \\ &= \left[\, \bar{C}_1 \; \bar{C}_2 \; \ldots \; \bar{C}_M \,\right] \end{aligned} \tag{5}$$

where we have introduced:

$$C_{k,k+1} = \left[\begin{array}{ccc|c} I_m & & & 0 \cdots 0 \\ & \ddots & & \vdots \; \ddots \; \vdots \\ & & I_m & 0 \cdots 0 \end{array}\right] \in \mathbb{R}^{m n_{c,(k,k+1)} \times \bar{m}} \tag{6}$$

i.e. each block row of $C_{k,k+1}$ corresponds to a common node in the scenario tree for scenario k and scenario $k+1$.

2.2 Dual decomposition

We introduce the dual variables $\lambda \in \mathbb{R}^{m(M-1)}$ corresponding to the non-anticipativity constraints (2b), and the dual variables $\mu_k \in \mathbb{R}^{n(N-1)}$ corresponding to the dynamics (2c), and define the partial Lagrange function:

$$\mathcal{L}(x,u,\mu,\lambda) = \sum_{k=1}^{M} V_k(x_k,u_k) + \lambda^T \sum_{k=1}^{M} \bar{C}_k u_k + \sum_{k=1}^{M} \mu_k^T \left(\bar{A}_k x_k + \bar{B}_k u_k - b_k\right) \quad (7)$$

where we have introduced the notation $\mu = [\mu_1^T \dots \mu_M^T]^T$ for notational convenience. Observe that the Lagrange function is seperable in the primal variables x and u and in the dual variables μ, i.e. we can express $\mathcal{L}(x,u,\lambda)$ as:

$$\mathcal{L}(x,u,\mu,\lambda) = \sum_{k=1}^{M} \mathcal{L}_k(x_k,u_k,\mu_k,\lambda) \quad (8)$$

where we have introduced:

$$\mathcal{L}_k(x_k,u_k,\mu_k,\lambda) = V_k(x_k,u_k) + \lambda^T \bar{C}_k u_k + \mu_k^T \left(\bar{A}_k x_k + \bar{B}_k u_k - b_k\right) \quad (9)$$

Due to the decomposable structure of $\mathcal{L}(x,u,\mu,\lambda)$, the Lagrange dual function $d(\lambda) = -\min_{(x,u)\in\mathcal{Z}} \mathcal{L}(x,u,\mu,\lambda)$, can be evaluated in parallel as:

$$d(\mu,\lambda) = -\sum_{k=1}^{M} \min_{(x_k,u_k)\in\mathcal{Z}_k} \mathcal{L}_k(x_k,u_k,\mu_k,\lambda) \quad (10)$$

where we have introduced the feasible sets $\mathcal{Z}_k = \{(x_k,u_k) : \underline{x}_k \le x_k \le \overline{x}_k,\ \underline{u}_k \le u_k \le \overline{u}_k\}$, and $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_M$.

Due to strict convexity of (2), $d(\mu,\lambda)$ is convex and continuously differentiable. The Hessian of $d(\mu,\lambda)$ is a piecewise constant matrix and changes with the active-set [8]. The non-smooth dual problem is then given by:

$$\min_{\mu,\lambda} d(\mu,\lambda) \quad (11)$$

from which the primal solution x^* and u^* to (2) can be recovered due to strong duality [5].

The gradient of the dual function is given by [2]:

$$\nabla d(\mu,\lambda) = \begin{bmatrix} r_1(x_1^*(\mu_1,\lambda), u_1^*(\mu_1,\lambda)) \\ \vdots \\ r_M(x_M^*(\mu_M,\lambda), u_M^*(\mu_1,\lambda)) \\ r(u^*(\mu,\lambda)) \end{bmatrix} \quad (12)$$

where $x_k^*(\mu_k,\lambda)$ and $u_k^*(\mu_k,\lambda)$ are solutions to the subproblems:

$$\begin{bmatrix} x_k^*(\mu_k,\lambda) \\ u_k^*(\mu_k,\lambda) \end{bmatrix} = \arg \min_{(x_k,u_k)\in\mathcal{Z}_k} \mathcal{L}_k(x_k,u_k,\mu_k,\lambda) \quad (13)$$

while r_k and r represent the residual of the dualized constraints, i.e.:

$$r_k(x_k, u_k) = -\bar{A}_k x_k(\mu_k, \lambda) - \bar{B}_k u_k(\mu_k, \lambda) + b_k \tag{14a}$$

$$r(u) = -\sum_{k=1}^{M} \bar{C}_k u_k(\mu_k, \lambda) \tag{14b}$$

Following directly from (12), the Hessian of the dual function takes the form of the following (permuted) block arrowhead matrix:

$$\nabla^2 d(\mu, \lambda) = \begin{bmatrix} \frac{\partial r_1^*}{\partial \mu_1} & & & \frac{\partial r_1^*}{\partial \lambda} \\ & \ddots & & \vdots \\ & & \frac{\partial r_M^*}{\partial \mu_M} & \frac{\partial r_M^*}{\partial \lambda} \\ \frac{\partial r^*}{\partial \mu_1} & \cdots & \frac{\partial r^*}{\partial \mu_M} & \frac{\partial r^*}{\partial \lambda} \end{bmatrix} \tag{15}$$

where we have omitted the arguments and used the notations $r_k^* = r_k(x_1^*(\mu_k, \lambda), u_k^*(\mu_k, \lambda))$ and $r^* = r(u^*(\mu, \lambda))$ for notational simplicity.

In this paper, we propose to solve (11) by updating the dual variables μ and λ according to:

$$\begin{bmatrix} \mu^+ \\ \lambda^+ \end{bmatrix} = \begin{bmatrix} \mu \\ \lambda \end{bmatrix} + t \begin{bmatrix} \Delta\mu \\ \Delta\lambda \end{bmatrix} \tag{16}$$

where $\Delta\mu$ and $\Delta\lambda$ is a solution to the regularized Newton system:

$$\left(\nabla^2 d(\mu, \lambda) + \bar{F}\right) \begin{bmatrix} \Delta\mu \\ \Delta\lambda \end{bmatrix} = -\nabla d(\mu, \lambda) \tag{17}$$

and the step size $t \in (0, 1]$ is chosen to enforce convergence. The regularization $\bar{F}$ is selected to enforce $\nabla^2 d(\mu, \lambda) + \bar{F} \succ 0$, whenever the dual Hessian is rank deficient. Specifically, this happens when the active local constraints together with the dualized constraints form linear dependencies [14]. In cases where LICQ is fulfilled for all compatible active sets, $\bar{F}$ can be omitted.

The proceeding is organized as follows. In Section 3, we describe an efficient way for calculating the dual gradient and the dual Hessian, whereas in Section 4, we propose an efficient method for solving the Newton system. The resulting optimization procedure is summarized in Section 5.

3 Calculating derivatives

In this section, we detail a method for calculating the dual gradient and the dual Hessian. Specifically, we propose a method for solving the subproblems (13), that are needed for the forming of the gradient, and a method for calculating the partial derivatives needed to form the Hessian.

3.1 Solving the subproblems

To evaluate the dual function, see (10), and its gradient, see (12), the optimal solutions $x_k^*(\mu_k, \lambda)$ and $u_k^*(\mu_k, \lambda)$ to the subproblems are needed. Because of the separability of the Lagrange function the calculations are separable and can be performed in parallel on a multiple core architecture. Additionally, as we shall see, the dualization of the non-anticipativity constraints and the dynamic equations, results in subproblems with trivial solutions.

Note that $x_k^*(\mu_k, \lambda)$ and $u_k^*(\mu_k, \lambda)$ are given as solutions to a QP of the following form:

$$\min_{x_k, u_k} \; V_k(x_k, u_k) + \mu_k^T \bar{A}_k x_k + \left(\lambda^T \bar{C}_k + \mu_k^T \bar{B}_k\right) u_k \tag{18a}$$

$$\text{s.t.} \; \underline{x}_k \leq x_k \leq \overline{x}_k \tag{18b}$$

$$\underline{u}_k \leq u_k \leq \overline{u}_k \tag{18c}$$

Since the matrices $\bar{Q}_k$ and $\bar{R}_k$ are diagonal, their eigenvectors are aligned with the coordinate axes, i.e. aligned with the box constraints. This implies that the optimal solution can, as described in [7], be found by a component-wise clipping of the unconstrained solution, i.e. $x_k^*(\mu_k, \lambda)$ and $u_k^*(\mu_k, \lambda)$ can be calculated as:

$$x_k^*(\mu_k, \lambda) = \text{mid}\left(\underline{x}_k, \overline{x}_k, -\bar{Q}_k^{-1}\left(\bar{q}_k + \bar{A}_k^T \mu_k\right)\right) \tag{19a}$$

$$u_k^*(\mu_k, \lambda) = \text{mid}\left(\underline{u}_k, \overline{u}_k, -\bar{R}_k^{-1}\left(\bar{r}_k + \bar{B}_k^T \mu_k + \bar{C}_k^T \lambda\right)\right) \tag{19b}$$

where $\text{mid}(a, b, c)$ is a vector containing the component-wise median of its three arguments.

The dominating computational cost for finding $x_k^*(\mu, \lambda)$ and $u_k^*(\mu, \lambda)$ is thus constituted by a banded matrix-vector multiplication which is negligable compared to solving the Newton system. As we shall see in the proceedings, this is instrumental in obtaining a dual Newton strategy with a fixed complexity per iteration.

3.2 Calculating the dual Hessian

In this subsection, we detail the calculation of the partial derivatives in the dual Hessian (15), i.e. the calculation of:

$$\frac{\partial r_k^*}{\partial \mu_k} = -\bar{A}_k \frac{\partial x_k^*(\mu_k, \lambda)}{\partial \mu_k} - \bar{B}_k \frac{\partial u_k^*(\mu_k, \lambda)}{\partial \mu_k} \tag{20a}$$

$$\frac{\partial r_k^*}{\partial \lambda} = -\bar{A}_k \frac{\partial x_k^*(\mu_k, \lambda)}{\partial \lambda} - \bar{B}_k \frac{\partial u_k^*(\mu_k, \lambda)}{\partial \lambda} \tag{20b}$$

$$\frac{\partial r^*}{\partial \mu_k} = -\bar{C}_k \frac{\partial u_k^*(\mu_k, \lambda)}{\partial \mu_k} \tag{20c}$$

$$\frac{\partial r^*}{\partial \lambda} = -\sum_{k=1}^{M} \bar{C}_k \frac{\partial u_k^*(\mu_k, \lambda)}{\partial \lambda} \tag{20d}$$

In the following, we denote the active constraints of (18b) and (18c) with:

$$\bar{D}_{\mathcal{A},k} x_k = d_{\mathcal{A},k} \tag{21a}$$
$$\bar{E}_{\mathcal{A},k} u_k = e_{\mathcal{A},k} \tag{21b}$$

Note that every row in $\bar{D}_{\mathcal{A},k}$ and $\bar{E}_{\mathcal{A},k}$ has only one non-zero element, namely 1 if the row corresponds to an active upper limit or -1 if the row corresponds to an active lower limit.

By introducing the dual variables y_k and z_k corresponding to (21a) and (21b) respectively, it follows from the optimality conditions of (18) that the following holds:

$$0 = \bar{Q}_k \frac{\partial x_k^*(\mu_k, \lambda)}{\partial \lambda} + \bar{D}_{\mathcal{A},k}^T \frac{\partial y_k^*(\mu_k, \lambda)}{\partial \lambda} \tag{22a}$$
$$0 = \bar{R}_k \frac{\partial u_k^*(\mu_k, \lambda)}{\partial \lambda} + \bar{C}_k^T + \bar{E}_k^T \frac{\partial z_k^*(\mu_k, \lambda)}{\partial \lambda} \tag{22b}$$
$$0 = \bar{D}_{\mathcal{A},k} \frac{\partial x_k^*(\mu_k, \lambda)}{\partial \lambda} \tag{22c}$$
$$0 = \bar{E}_{\mathcal{A},k} \frac{\partial u_k^*(\mu_k, \lambda)}{\partial \lambda} \tag{22d}$$

By solving (22) for $\frac{\partial x_k^*(\mu_k,\lambda)}{\partial \lambda}$ and $\frac{\partial u_k^*(\mu_k,\lambda)}{\partial \lambda}$ we obtain:

$$\frac{\partial x_k^*(\mu_k, \lambda)}{\partial \lambda} = 0 \tag{23a}$$
$$\frac{\partial u_k^*(\mu_k, \lambda)}{\partial \lambda} = -\bar{R}_{\mathcal{I},k}^{-1} \bar{C}_k^T \tag{23b}$$

where we have introduced $\bar{R}_{\mathcal{I},k}^{-1} = \bar{R}_k^{-1}(I - \bar{E}_{\mathcal{A},k}^T \bar{E}_{\mathcal{A},k})$. Observe that $\bar{R}_{\mathcal{I},k}^{-1}$ is identical to $\bar{R}_k^{-1}$, except that every diagonal element corresponding to an active constraint is replaced by a zero.

Let us now focus on the computation of $\frac{\partial x_k^*(\mu_k,\lambda)}{\partial \mu_k}$ and $\frac{\partial u_k^*(\mu_k,\lambda)}{\partial \mu_k}$. Similarly to (22), we obtain the following linear system:

$$0 = \bar{Q}_k \frac{\partial x_k^*(\mu_k, \lambda)}{\partial \mu_k} + \bar{A}_k^T + \bar{D}_{\mathcal{A},k}^T \frac{\partial y_k^*(\mu_k, \lambda)}{\partial \mu_k} \tag{24a}$$
$$0 = \bar{R}_k \frac{\partial u_k^*(\mu_k, \lambda)}{\partial \mu_k} + \bar{B}_k^T + \bar{E}_k^T \frac{\partial z_k^*(\mu_k, \lambda)}{\partial \mu_k} \tag{24b}$$
$$0 = \bar{D}_{\mathcal{A},k} \frac{\partial x_k^*(\mu_k, \lambda)}{\partial \mu_k} \tag{24c}$$
$$0 = \bar{E}_{\mathcal{A},k} \frac{\partial u_k^*(\mu_k, \lambda)}{\partial \mu_k} \tag{24d}$$

Using block elimination, it can be verified that:

$$\frac{\partial x_k^*(\mu_k, \lambda)}{\partial \mu} = -\bar{Q}_{\mathcal{I},k}^{-1} \bar{A}_k^T \tag{25a}$$
$$\frac{\partial u_k^*(\mu_k, \lambda)}{\partial \mu_k} = -\bar{R}_{\mathcal{I},k}^{-1} \bar{B}_k^T \tag{25b}$$

where we have introduced the notation $\bar{Q}_{\mathcal{I},k}^{-1} = \bar{Q}_k^{-1}(I - \bar{D}_{\mathcal{A},k}^T \bar{D}_{\mathcal{A},k})$. Note that $\bar{Q}_{\mathcal{I},k}^{-1}$ is identical to $\bar{Q}_k^{-1}$, except that every diagonal element corresponding to an active constraint is replaced by a zero.

Finally, by using (23) and (25) the expressions in (20) can be simplified as:

$$\frac{\partial r_k^*}{\partial \mu_k} = \bar{A}_k \bar{Q}_{\mathcal{I},k}^{-1} \bar{A}_k^T + \bar{B}_k \bar{R}_{\mathcal{I},k}^{-1} \bar{B}_k^T \tag{26a}$$

$$\frac{\partial r_k^*}{\partial \lambda} = \bar{B}_k \bar{R}_{\mathcal{I},k}^{-1} \bar{C}_k^T \tag{26b}$$

$$\frac{\partial r^*}{\partial \mu_k} = \bar{C}_k \bar{R}_{\mathcal{I},k}^{-1} \bar{B}_k^T \tag{26c}$$

$$\frac{\partial r^*}{\partial \lambda} = \sum_{k=1}^{M} \bar{C}_k \bar{R}_{\mathcal{I},k}^{-1} \bar{C}_k^T \tag{26d}$$

This implies that the dual Hessian can be formed cheaply at the cost of a banded matrix-matrix multiplication once the subproblems are solved and the active set is known. In [7], a similar expression is used for the dual Hessian in the context of standard MPC.

4 Efficient solution of the Newton system

To achieve an efficient implementation, it is crucial to exploit the structure of the Newton system in the computation of the search direction. Additionally, since the dual Hessian can be singular, a regularization strategy is needed in order to solve (17). In this section, we aim at addressing these issues.

4.1 Alternative Newton system

To exploit the sparsity structure of (15), block elimination can be used to obtain alternate formulations of the search direction computation. For problems of the form (2), we propose to exploit the sparsity structure and to enable parallel computations by eliminating $\Delta\mu_k$ from (17). In this case, the regularization matrix $\bar{F}$ can be calculated implicitly in a parallel fashion. The resulting formulation can be expressed as:

$$(J + F)\Delta\lambda = -r + \sum_{k=1}^{M} \frac{\partial r}{\partial \mu_k} \left(\frac{\partial r_k}{\partial \mu_k} + F_k \right)^{-1} r_k \tag{27a}$$

$$\left(\frac{\partial r_k}{\partial \mu_k} + F_k \right) \Delta\mu_k = -r_k - \frac{\partial r_k}{\partial \lambda} \Delta\lambda, \quad k = 1, \ldots, M \tag{27b}$$

where we have introduced:

$$J = \frac{\partial r}{\partial \lambda} - \sum_{k=1}^{M} \frac{\partial r}{\partial \mu_k} \left(\frac{\partial r_k}{\partial \mu_k} + F_k \right)^{-1} \frac{\partial r_k}{\partial \lambda} \tag{28}$$

and $F_k \in \mathbb{S}_+^{nN}$ and $F \in \mathbb{S}_+^p$ are chosen to enforce that $\frac{\partial r_k}{\partial \mu_k} + F_k \succ 0$ and $J + F \succ 0$.

We can then first find $\Delta\lambda$ by solving (27a), and then use the result to find $\Delta\mu_k$. Observe that (27b) is trivially decomposable and can be solved in parallel on M CPUs.

4.2 Regularization

To guarantee that the search direction, $\Delta\mu$ and $\Delta\lambda$, is a descent direction, the regularized dual Hessian has to be positive definite. In the following lemma, we show that the formulation (27) results in search directions that are descent directions. This implies that the regularization matrix $\bar{F}$ can be calculated implicitly and partly in parallel without explicitly forming the dual Hessian.

Lemma 1 *If the conditions:*

1. $\frac{\partial r_k}{\partial \mu_k} + F_k \succ 0$, *for* $k = 1, \ldots, M$
2. $J + F \succ 0$

hold, then the regularized dual Hessian is positive definite.

Proof Recall that a Hermitian matrix is positive definite if and only if it has a uniquely defined Cholesky factor [10]. By restricting $\nabla^2 d(\mu, \lambda) + \bar{F}$ to have the same sparsity structure as the dual Hessian, we introduce the Cholesky factor $\bar{L}$, such that $\nabla^2 d(\mu, \lambda) + \bar{F} = \bar{L}\bar{L}^T$, i.e.:

$$\bar{L} = \begin{bmatrix} \bar{L}_{1,1} & & & \\ & \ddots & & \\ & & \bar{L}_{M,M} & \\ \bar{L}_{M+1,1} & \ldots & \bar{L}_{M+1,M} & \bar{L}_{M+1,M+1} \end{bmatrix} \tag{29}$$

where the lower triangular blocks $\bar{L}_{k,k} \in \mathbb{R}^{nN \times nN}$ and the rectangular blocks $\bar{L}_{M+1,k} \in \mathbb{R}^{p \times nN}$, can be calculated according to:

$$\bar{L}_{k,k}\bar{L}_{k,k}^T = \frac{\partial r_k}{\partial \mu_k} + F_k, \quad k = 1, \ldots, M \tag{30a}$$

$$\bar{L}_{M-1,k}\bar{L}_{k,k}^T = \frac{\partial r}{\partial \mu_k}, \quad k = 1, \ldots, M \tag{30b}$$

$$\bar{L}_{M+1,M+1}\bar{L}_{M+1,M+1}^T = \frac{\partial r}{\partial \lambda} - \sum_{k=1}^{M} \bar{L}_{M+1,k}\bar{L}_{M+1,k}^T + F \tag{30c}$$

For uniqueness of the of the Cholesky factor, observe that F_k and F must be chosen to enforce that the right hand sides of (30a) and (30c) are positive definite. Observe that positive definiteness of the right hand side of (30a) is identical to condition 1. Moreover, using (30a) and (30b), we can express (30c) as:

$$\bar{L}_{M+1,M+1}\bar{L}_{M+1,M+1}^T = J + F \tag{31}$$

Hence, positive definiteness of the right hand side of (30c), is identical to condition 2.

4.3 Finding search directions $\Delta\mu_k$

In this subsection, we detail the calculation of the search directions $\Delta\mu_k$, i.e. a solution method for (27b).

First, let us introduce the following notation:

$$\Lambda_k = \frac{\partial r_k}{\partial \mu_k} + F_k = \bar{A}_k \bar{Q}_{\mathcal{I},k}^{-1} \bar{A}_k^T + \bar{B}_k \bar{R}_{\mathcal{I},k}^{-1} \bar{B}_k^T + F_k \tag{32}$$

We observe that if F_k is restricted to have the same sparsity structure as $\frac{\partial r_k}{\partial \mu_k}$, then Λ_k has a block-tridiagonal structure , i.e.:

$$\Lambda_k = \begin{bmatrix} \Lambda_{k,(1,1)} & \Lambda_{k,(1,2)} & & \\ \Lambda_{k,(1,2)}^T & \Lambda_{k,(2,2)} & \ddots & \\ & \ddots & \ddots & \Lambda_{k,(N-1,N)} \\ & & \Lambda_{k,(N-1,N)}^T & \Lambda_{k,(N,N)} \end{bmatrix} \tag{33}$$

with blocks $\Lambda_{k,(i,j)} \in \mathbb{R}^{n \times n}$. Consequently, the Cholesky factor L_k of Λ_k which yield $\Lambda_k = L_k L_k^T$, is block-lower bidiagonal:

$$L_k = \begin{bmatrix} L_{k,(1,1)} & & & \\ L_{k,(2,1)} & L_{k,(2,2)} & & \\ & \ddots & \ddots & \\ & & L_{k,(N,N-1)} & L_{k,(N,N)} \end{bmatrix} \tag{34}$$

with blocks $L_{k,(i,j)} \in \mathbb{R}^{n \times n}$, which can be calculated using the following block-wise procedure:

$$\Lambda_{k,(1,1)} = L_{k,(1,1)} L_{k,(1,1)}^T \tag{35a}$$

$$\Lambda_{k,(i,i+1)} = L_{k,(i,i)} L_{k,(i+1,i)}^T, \quad i = 1, \ldots, N \tag{35b}$$

$$\Lambda_{k,(i,i)} - L_{k,(i,i-1)} L_{k,(i,i-1)}^T = L_{k,(i,i)} L_{k,(i,i)}^T, \; i = 2, \ldots, N \tag{35c}$$

Using L_k, the search direction $\Delta\mu_k$ can be calculated from (27b) via a sparse forward and backward substitution. Note that the calculation and factorization of Λ_k and the calculation of $\Delta\mu_k$ for $k = 1, \ldots, M$ can be performed in a parallel fashion on M CPUs.

4.4 Finding the search direction $\Delta\lambda$

The search direction $\Delta\lambda$ is provided by the linear system (27a). In this subsection, we establish the sparsity structure of J, and propose a method for solving (27a). Let us start with detailing the sparsity structure of J. Using (26) and (28), we write J as:

$$J = \sum_{k=1}^{M} \bar{C}_k \left(\bar{R}_{\mathcal{I},k}^{-1} - \bar{R}_{\mathcal{I},k}^{-1} \bar{B}_k^T \Lambda_k^{-1} \bar{B}_k \bar{R}_{\mathcal{I},k}^{-1} \right) \bar{C}_k^T \tag{36}$$

Using the structure of $\bar{C}_k$ detailed in (5), we note that J is block tridiagonal:

$$J = \begin{bmatrix} J_{1,1} & J_{1,2} & & \\ J_{1,2} & J_{2,2} & \ddots & \\ & \ddots & \ddots & J_{M-2,M-1} \\ & & J_{M-2,M-1} & J_{M-1,M-1} \end{bmatrix} \tag{37}$$

with blocks $J_{k,k} \in \mathbb{S}^{mn_{c,(k,k+1)}}$ and $J_{k,k+1} \in \mathbb{R}^{mn_{c,(k+1,k+2)} \times mn_{c,(k,k+1)}}$ given by:

$$J_{k,k} = C_{k,k+1}(K_k + K_{k+1})C_{k,k+1}^T \tag{38a}$$
$$J_{k-1,k} = -C_{k-1,k}K_k C_{k,k+1}^T \tag{38b}$$

where we have introduced:

$$K_k = \bar{R}_{\mathcal{I},k}^{-1} - \bar{R}_{\mathcal{I},k}^{-1}\bar{B}_k^T \Lambda_k^{-1} \bar{B}_k \bar{R}_{\mathcal{I},k}^{-1} \tag{39}$$

Note that the size of the blocks in the block-tridiagonal structure of J are inherited from the structure of the scenario tree. More specifically, the size of $J_{k,k}$ is determined by the number of common nodes between scenarios k and $k+1$, whereas the size of $J_{k,k+1}$ is determined by the number of common nodes between scenario k, $k+1$ and scenario $k+1$, $k+2$.

If F is restricted to have the same sparsity structure as J the Cholesky factor L, yielding $J + F = LL^T$, is block lower bidiagonal:

$$L = \begin{bmatrix} L_{1,1} & & & \\ L_{2,1} & L_{2,2} & & \\ & \ddots & \ddots & \\ & & L_{M-1,M-2} & L_{M-1,M-1} \end{bmatrix} \tag{40}$$

and can be efficiently calculated according to:

$$J_{1,1} + F_{1,1} = L_{1,1}L_{1,1}^T \tag{41a}$$
$$J_{k,k+1} + F_{k,k+1} = L_{k,k}L_{k+1,k}^T, \quad k = 1, \ldots, M-1 \tag{41b}$$
$$J_{k,k} + F_{k,k} - L_{k,k-1}L_{k,k-1}^T = L_{k,k}L_{k,k}^T, \; k = 2, \ldots, M-1 \tag{41c}$$

Using L, the search direction $\Delta\lambda$ can be calculated from (27a) via a sparse forward and backward substitution.

4.5 Forming matrix J

In this subsection, we propose a parallelizable method for forming matrix J, where we put the emphasis on reusing components that are available from

previous computations. First, let us introduce the following block partitioning of K_k:

$$K_k = \begin{bmatrix} K_{k,(1,1)} & \cdots & K_{k,(N,1)}^T \\ \vdots & \ddots & \vdots \\ K_{k,(N,1)} & \cdots & K_{k,(N,N)} \end{bmatrix} \tag{42}$$

with blocks $K_{k,(i,j)} \in \mathbb{R}^{m \times m}$, and the notation:

$$K_k^{N_1,N_2} = \begin{bmatrix} K_{k,(1,1)} & \cdots & K_{k,(N_2,1)}^T \\ \vdots & \ddots & \vdots \\ K_{k,(N_1,1)} & \cdots & K_{k,(N_1,N_2)} \end{bmatrix} \tag{43}$$

for the $N_1 \times N_2$ most top-left blocks of matrix K_k. Due to the structure of $C_{k,k+1}$, see (6), the expressions in (38) can now be simplified as:

$$J_{k,k} = K_k^{n_{c,(k,k+1)},n_{c,(k,k+1)}} + K_{k+1}^{n_{c,(k,k+1)},n_{c,(k,k+1)}} \tag{44a}$$

$$J_{k-1,k} = -K_k^{n_{c,(k-1,k)},n_{c,(k,k+1)}} \tag{44b}$$

Accordingly, our concern is to calculate a part of K_k for each scenario, where the size of the part that needs to be computed is governed by the number of common nodes in the scenario tree for scenario k and its neighboring scenarios. In the following, we propose a method for calculating top-left parts of K_k.

Let us first make the observation that we can express K_k as:

$$K_k = \bar{R}_{\mathcal{I},k}^{-1} - \bar{Z}_k^T \Lambda_k^{-1} \bar{Z}_k \tag{45}$$

where we have introduced:

$$\bar{Z}_k = \bar{B}_k \bar{R}_{\mathcal{I},k}^{-1} \tag{46}$$

Let us now consider the calculation of the term $\bar{Z}_k^T \Lambda_k^{-1} \bar{Z}_k$. Due to symmetry, we can find a rectangular factor U_k, such that $\bar{Z}_k^T \Lambda_k^{-1} \bar{Z}_k = U_k^T U_k$, where we introduce the following block partitioning of U_k:

$$U_k = \begin{bmatrix} U_{k,(1,1)} & \cdots & U_{k,(1,N)} \\ \vdots & \ddots & \vdots \\ U_{k,(N,1)} & \cdots & U_{k,(N,N)} \end{bmatrix} \tag{47}$$

with blocks $U_{k,(i,j)} \in \mathbb{R}^{m \times m}$. By using this partitioning, we can express $K_{k,(i,i)}$ and $K_{k,(i,j)}$ as:

$$K_{k,(i,i)} = R_{\mathcal{I},k,i}^{-1} - U_{k,(1,i)}^T U_{k,(1,i)} - \cdots - U_{k,(N,i)}^T U_{k,(N,i)} \tag{48a}$$

$$K_{k,(i,j)} = -U_{k,(1,i)}^T U_{k,(1,j)} - \cdots - U_{k,(N,i)}^T U_{k,(N,j)} \tag{48b}$$

where $R_{\mathcal{I},k,i}^{-1} \in \mathbb{R}^{m \times m}$ denotes the block of $\bar{R}_{\mathcal{I},k}^{-1}$ corresponding to time stage $i+1$. This implies that we need to compute the first $n_k = \max\{n_{c,(k-1,k)}, n_{c,(k,k+1)}\}$ block columns of U_k, in the following denoted as $U_k^{n_k}$, in order to assemble

the required blocks $K_{k,(i,j)}$. Next, we propose to compute $U_k^{n_k}$ by reusing the Cholesky factor L_k in (34).

Observe that the factor U_k can be calculated according to:

$$L_k U_k = \bar{Z}_k \tag{49}$$

This implies that $U_k^{n_k}$ can be computed by using only the corresponding part of $\bar{Z}_k$, i.e. by solving:

$$L_k U_k^{n_k} = Z_k^{n_k} \tag{50}$$

where $Z_k^{n_k} \in \mathbb{R}^{nN \times mn_k}$ denotes the first mn_k columns of $\bar{Z}_k$.

The computational cost of finding J is accordingly constituted by M matrix forward substitutions, to calculate $U_k^{n_k}$, and M matrix-matrix multiplications, to calculate the required parts of K_k. The computations are, however, trivially decomposable and can be performed in parallel on M CPUs.

5 A dual Newton method with a fixed complexity per iteration

In this section, we provide a summary of the proposed dual Newton strategy and suggest algorithmic details to enhance its performance.

The method is summarized in Algorithm 1. Note that each iteration consists of a, possibly damped, Newton step on the dual variables, i.e. the calculation of the search directions $\Delta\mu$ and $\Delta\lambda$ and an appropriately chosen step size t. The calculation of the search directions can be performed to a large extent in parallel as described in Section 3 and 4.

Algorithm 1: Dual Newton strategy

Input: w_k, λ, τ
while *No convergence* **do**
  Solve subsystems to obtain $x_k^*(\mu_k, \lambda)$ and $u_k^*(\mu_k, \lambda)$
  Calculate the blocks in $\nabla^2 d(\mu, \lambda)$ using (26)
  Solve Newton system to obtain $\Delta\mu$ and $\Delta\lambda$
  Compute an appropriate step size t
  Update dual variables according to (16)
end

5.1 Choice of step size

Because of the non-smooth dual function, a globalization strategy is needed to ensure convergence of the dual Newton strategy. Several line-search strategies have been proposed [8], with the common property that they require the evaluation of the dual function for candidate step sizes $t \in (0, 1]$, i.e. the evaluation of $d(\mu + t\Delta\mu, \lambda + t\Delta\lambda)$. Since the dual function can be evaluated by performing only small matrix-vector multiplications, which are negligible compared to

solving the large Newton system, the computational cost for performing one iteration of Algorithm 1 is essentially constant.

In this paper, we select t based on an Armijo line search with backtracking. Specifically, for every search direction $\Delta\mu$ and $\Delta\lambda$, we initialize the step size as $t = 1$, and backtrack according to:

$$t^+ = \beta t \tag{51}$$

until the following criterion is fulfilled:

$$d(\mu + t\Delta\mu, \lambda + t\Delta\lambda) \leq d(\mu, \lambda) + \sigma \nabla d(\mu, \lambda)^T \begin{bmatrix} \Delta\mu \\ \Delta\lambda \end{bmatrix} \tag{52}$$

for constants $\sigma, \beta \in (0, 1)$. Note that because of the low computational cost for evaluating the dual function, a careful line search can be afforded, i.e. large constants σ and β can be used. Additionally, observe that due to convexity of $d(\mu, \lambda)$, a step size $t \in (0, 1]$ can always be found such that (52) is fulfilled.

Finally, we note that the line search strategy is parallelizable since the evaluation of the dual function can be performed in parallel on M CPUs. In this context the communication overhead resulting from the parallelization can be reduced by evaluating several candidate step sizes at once.

5.2 Elimination of redundant constraints

In the context of dual Newton strategies, the Hessian of the dual function is rank deficient if the active constraints (2d) and (2e) together with the dualized constraints (2b) and (2c) are linearly dependent, leading to an inconsistent Newton system [14]. This implies that the need for regularization can be reduced by eliminating redundant constraints.

For the multi-stage MPC problem (2), linear dependencies occur if inequality constraints become active for scenarios participating in the same node. In consequence, for every node in the scenario-tree, we only enforce an inequality constraint on one scenario participating in that node. The inequality constraints are thus enforced implicitly on the other scenarios via the dynamics and the non-anticipativity constraints. For a detailed description see [12].

5.3 Warm-starting

One of the key advantages of using the dual Newton strategy in the context of MPC is its capability to exploit the similarity between two subsequent QPs. Specifically, if the method is provided with an initialization of the dual variables which corresponds to the correct active set, a one step convergence follows [8].

An often practically efficient strategy for standard MPC, is to initialize the method by using a time shift of the dual solution of the previous problem instance. Due to the tree structure of a multi-stage MPC problem, a time

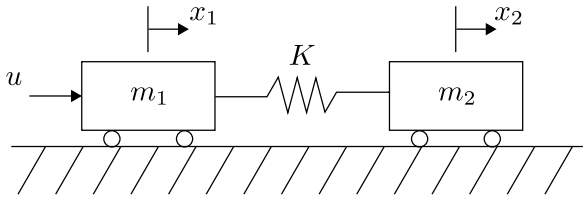

Fig. 2: Illustration of two-mass-spring system.

shift of this kind can be performed in several different ways. However, we have not found a shifting strategy that is consistently performing better than a warm-start where the problem is initialized at the non-shifted dual solution of the previous problem instance. For this reason, we only consider a non-shifted warm-start in the numerical experiments.

6 Numerical experiments

To study the properties of the presented dual Newton scheme we consider a benchmark example of a two-mass-spring system [25], [13], as illustrated in Figure 2. The two carts have the same mass $m_1 = m_2 = 1\,\text{kg}$ while the spring constant K is uncertain, However, we know that $K \in [K_{\min}, K_{\max}]$ where $K_{\min} = 0.5\,\text{N/m}$ and $K_{\max} = 10\,\text{N/m}$. The linear model that describes the system in continuous time is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -K/m_1 & K/m_1 & 0 & 0 \\ K/m_2 & -K/m_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/m_1 \\ 0 \end{bmatrix} u \tag{53}$$

where x_1 and x_2 denote the relative position of the two carts with respect to their initial position, x_3 and x_4 denote the velocities of the carts and u is the force applied on the active cart, i.e. on cart number 1 as depicted in Figure 2. We assume that full state measurements are available and define the following control objectives:

1. Track a unit-step command on the position x_2.
2. Keep control actions in the range $|u| \leq 1\,\text{N}$.
3. Allow a maximum of 20% overshoot on x_2.

To fulfill these requirements under the given parameter uncertainty we use the multi-stage MPC framework with prediction horizon $N = 50$, robust horizon $N_r = 2$ and $m_d = 3$ realizations of the uncertainty, namely $K \in \{0.5, 5.25, 10\}$, i.e. we consider $m_d^{N_r} = 9$ scenarios. The continuous time dynamics in (53) are discretized using matrix exponentials, assuming piecewise constant controls in intervals of $T_s = 0.1\,\text{s}$. The steady state point to be tracked is $x_{\text{ref}} = [1,\, 1,\, 0,\, 0]^T, u_{\text{ref}} = 0$ and the initial condition $x_0 = [0,\, 0,\, 0,\, 0]^T$. Finally, the quadratic weights in the objective function are chosen constant and

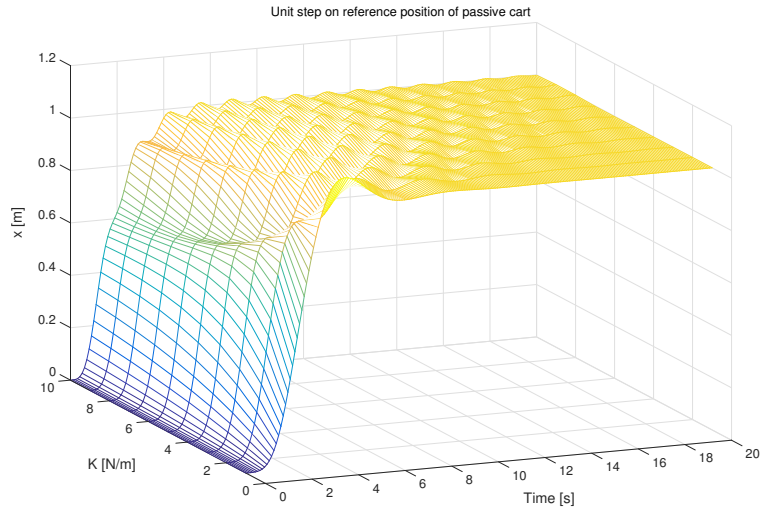

Fig. 3: Position of passive cart in closed loop for the whole range of spring constant values in the simulation model.

equal to:

$$Q_{k,i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \ \forall k, i \neq N \tag{54a}$$

$$Q_{k,N} = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \tag{54b}$$

$$R_{k,i} = 1, \ \forall k, i \tag{54c}$$

To validate the robustness of the adopted control scheme, we first run a series of closed-loop simulations with different values of the spring constant K_{nom} in the simulation model. The closed-loop trajectories of the cart position x_2 for $K_{\text{nom}} \in \{1, 2 \ldots, 10\}$ are shown in Figure 3. The stopping criterion is $\|\nabla d(\mu, \lambda)\|_\infty < 10^{-4}$. The predicted optimal state and control trajectories for one MPC step for a non-trivial problem instance with $K_{\text{nom}} = 2$ are depicted in Figure 4. Note that the control variables for the scenarios are identical on the robust horizon and that the state and control bounds are respected.

In the experiments we use a Levenberg-Marquardt regularization, i.e. we select $F = \delta I$ whenever J is singular and $F_k = \delta I$ whenever $\frac{\partial r_k}{\partial \mu_k}$ is singular, where we use $\delta = 10^{-8}$. Table 1 summarizes the maximum and average number of iterations in a closed-loop simulation with $K_{\text{nom}} = 10$, with and without the redundant constraints as described in Section 5.2, for a cold-started and warm-started scheme as described in Section 5.3. Note that the performance is

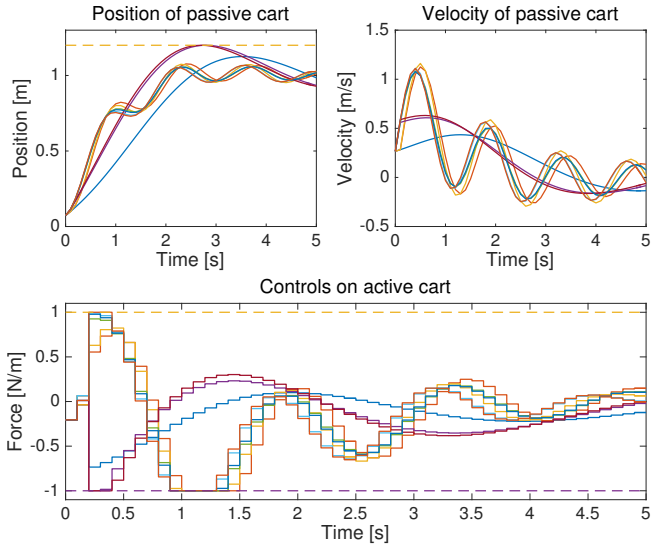

Fig. 4: Optimized trajectories for x_2, x_4 and u in one MPC problem instance.

Table 1: Dual Newton iterations in closed-loop.

Elim. of redund. const.	Initialization	Max. no. it.	Avg. no. it.
Yes	Cold start at zero	36	12
Yes	Warm-start	16	5.3
No	Cold start at zero	39	12
No	Warm-start	17	5.5

significantly improved by employing a warm-started scheme whereas a modest improvement is achieved by eliminating the redundant constraints. The average number of iterations in Table 1 are calculated over the non-trivial part of the simulation, i.e., until no inequality constraints are active at the solution.

To conclude, we visualize some essential sparsity patterns based on the benchmark example of this section. To make the structures more visible we use a shorter prediction horizon of $N = 5$ here. Figure 5 depicts the structure of the dual Hessian $\nabla^2 d(\mu, \lambda)$ which is in agreement with (15). The block tridiagonal matrices Λ_k and J, which allow for efficient factorizations, are shown in Figure 6.

7 Conclusions

In this paper, we present a dual Newton strategy for multi-stage MPC. To achieve a fixed computational complexity per iteration, we propose to dualize both the dynamic constraints and the non-anticipativity constraints. As a result, the dual function can be evaluated very cheaply, and the gradient

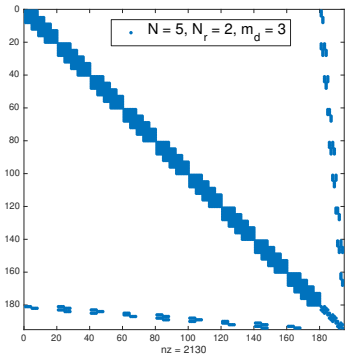

Fig. 5: Sparsity pattern of dual Hessian.

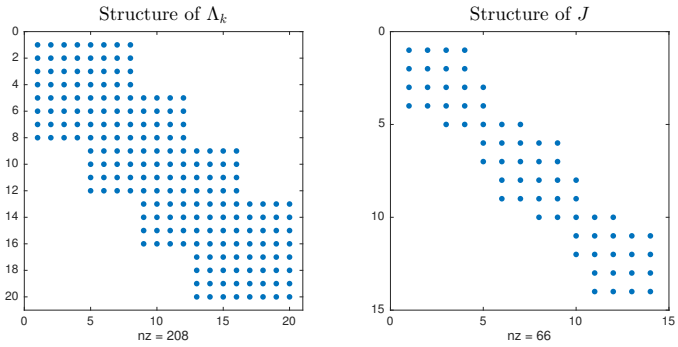

Fig. 6: Sparsity patterns of matrices Λ_k and J.

and Hessian of the dual function can be formed at a low computational cost. Additionally, the dual Newton system can be reformulated into several small and highly structured linear systems that to a large extent can be solved in parallel. As a consequence, the resulting method supports a large degree of parallelism.

References

1. Bernardini, D., Bemporad, A.: Scenario-based model predictive control of stochastic constrained linear systems. In: IEEE Conference on Decision and Control (2009)
2. Bertsekas, D., Tsitsiklis, J.N.: Parallel and distributed computation: Numerical methods. Prentice Hall (1989)
3. Birge, J.: Decomposition and Partitioning Methods for Multistage Stochastic Linear Programs. Operations Research **33**, 989–1007 (1985)
4. Birge, J.R., Holmes, D.F.: Efficient Solution of Two-Stage Stochastic Linear Programs Using Interior Point Methods. Computational Optimization and Applications **1**, 245–276 (1992)
5. Boyd, S., Vandenberghe, L.: Convex Optimization. University Press, Cambridge (2004)
6. Campo, P.J., Morari, M.: Robust model predictive control. In: American Control Conference, pp. 1021–1026 (1987)

7. Ferreau, H., Kozma, A., Diehl, M.: A parallel active-set strategy to solve sparse parametric quadratic programs arising in MPC. In: IFAC Nonlinear Model Predictive Control Conference (2012)
8. Frasch, J.V., Sager, S., Diehl, M.: A Parallel Quadratic Programming Method for Dynamic Optimization Problems. Mathematical Programming Computation **7**, 289–329 (2015)
9. Hansson, A.: A primal-dual interior-point method for robust optimal control of linear discrete-time systems. IEEE Transactions on Automatic Control **45**, 1639–1655 (2000)
10. Horn, R.A., Johnson, C.R.: Matrix Analysis - 2nd ed. Cambridge University Press (2013)
11. Jessup, E., Yang, D., Zenios, S.: Parallel factorization of structured matrices arising in stochastic programming. SIAM Journal on Optimization **4**, 833–846 (1994)
12. Klintberg., E., Dahl., J., Fredriksson., J., Gros., S.: An improved dual Newton strategy for scenario-tree MPC. In: IEEE Conference on Decision and Control (submitted) (2016)
13. Kothare, M., Balakrishnan, V., Morari, M.: Robust constrained model predictive control using linear matrix inequalities. Automatica **32**, 1361–1379 (1996)
14. Kozma, A., Klintberg, E., Gros, S., Diehl, M.: An improved distributed dual Newton-CG method for convex quadratic programming problems. In: American Control Conference (2014)
15. Leidereiter, C., Potschka, A., Bock, H.G.: Dual decomposition for QPs in scenario tree NMPC. In: European Control Conference (2015)
16. Lucia, S., Andersson, J., Brandt, H., Diehl, M., Engell, S.: Handling uncertainty in economic model predictive control: A comparative case study. Journal of Process Control **24**, 1247–1259 (2014)
17. Lucia, S., Finkler, T., Basak, B., Engell, S.: A new Robust NMPC Scheme and its Application to a Semi-batch Reactor Example. In: IFAC International Symposium on Advanced Control of Chemical Processes (2012)
18. Maiworm, M., Bathge, T., Findeisen, R.: Scenario-based model predictive control: Recursive feasibility and stability. In: IFAC Symposium on Advanced Control of Chemical Processes (2015)
19. Marti, R., Lucia, S., Sarabia, D., Paulen, R., Engell, S.: Improving scenario decomposition algorithms for robust nonlinear model predictive control. Computers and Chemical Engineering **79**, 30–45 (2015)
20. Mayne, D., Rakovic, S., Findeisen, R., Allgöwer, F.: Robust output feedback model predictive control of constrained linear systems. Automatica **42**, 1217–1222 (2006)
21. Mayne, D., Seron, M., Rakovic, S.: Robust model predictive control of constrained linear systems with bounded disturbances. Automatica **41**, 219–224 (2005)
22. Mulvey, J., Ruszczyński, A.: A New Scenario Decomposition Method for Large-Scale Stochastic Optimization. Operations Research **43**, 477–490 (1995)
23. Munoz de la Pena, D., Bemporad, A., Alamo, T.: Stochastic programming applied to model predictive control. In: IEEE Conference on Decision and Control (CDC) (2005)
24. Steinbach, M.C.: Tree-Sparse Convex Programs. Mathematical Methods of Operations Research **56**(3), 347–376 (2002)
25. Wie., B., Bernstein., D.: Benchmark Problems for Robust Control Design. Journal of Guidance Control and Dynamics **15**, 1057–1059 (1992)
26. Zeilinger, M., Raimondo, D., Domahidi, A., Morari, M., Jones, C.: On real-time robust model predictive control. Automatica **50**, 683–694 (2014)