Complementary Condensing for the Direct Multiple Shooting Method

Christian Kirches, Hans Georg Bock, Johannes P. Schlöder, Sebastian Sager

Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg
Im Neuenheimer Feld 368, 69120 Heidelberg, Germany

Summary. In this contribution we address the efficient solution of optimal control problems of dynamic processes with many controls. Such problems typically arise from the convexification of integer control decisions. We treat this problem class using the direct multiple shooting method to discretize the optimal control problem. The resulting nonlinear problems are solved using an SQP method. Concerning the solution of the quadratic subproblems we present a factorization of the QP's KKT system, based on a combined null–space range–space approach exploiting the problem's block sparse structure. We demonstrate the merit of this approach for a vehicle control problem in which the integer gear decision is convexified.

1 Introduction

Mixed–integer optimal control problems (MIOCPs) in ordinary differential equations (ODEs) have a high potential for optimization. A typical example is the choice of gears in transport [6, 8, 9, 14, 19].

Direct methods, in particular *all–at–once approaches*, [3, 2], have become the methods of choice for most practical OCPs. The drawback of direct methods with binary control functions is that they lead to high–dimensional vectors of binary variables. Because of the exponentially growing complexity of the problem, techniques from mixed–integer nonlinear programming will work only for small instances [20].

In past contributions [9, 12, 15, 13] we proposed to use an *outer convexification* with respect to the binary controls, which has several main advantages over standard formulations or convexifications, cf. [12, 13]. In an SQP framework for the solution of the discretized MICOP, the outer convexification approach results in QPs with many control parameters. Classical methods [3] for exploiting the block sparse structure of the discretized OCP leave room for improvement.

In [16, 17], structured interior point methods for solving QP subproblems arising in SQP methods for the solution of discretized nonlinear OCPs are studied. A family of block structured factorizations for the arising KKT systems is presented. Extensions to tree–sparse convex programs can be found in [18].

In this contribution we present an alternative approach at solving these QPs arising from outer convexification of MIOCPs, showing that a certain factorization from

[16] ideally lends itself to the case of many control parameters. We employ this factorization for the first time inside an active–set method. Comparisons of run times and complexity to classical condensing methods are presented.

2 Direct Multiple Shooting for Optimal Control

2.1 Optimal Control Problem Formulation

In this section we describe the direct multiple shooting method [3] as an efficient tool for the discretization and parameterization of a broad class of OCPs. We consider the following general class (1) of optimal control problems

$$\min_{x(\cdot),u(\cdot)} \quad l(x(\cdot),u(\cdot)) \tag{1a}$$

$$\text{s.t.} \quad \dot{x}(t) = f(t,x(t),u(t)) \qquad \forall t \in \mathscr{T} \tag{1b}$$

$$0 \leq c(t,x(t),u(t)) \qquad \forall t \in \mathscr{T} \tag{1c}$$

$$0 \leqq r(t_i,x(t_i)) \qquad 0 \leq i \leq m \tag{1d}$$

in which we strive to minimize objective function $l(\cdot)$ depending on the trajectory $x(\cdot)$ of a dynamic process described in terms of a system f of ordinary differential equations the time horizon $\mathscr{T} := [t_0,t_\mathrm{f}] \subset \mathbb{R}$, and governed by a control trajectory $u(\cdot)$ subject to optimization. The process trajectory $x(\cdot)$ and the control trajectory $u(\cdot)$ shall satisfy certain inequality path constraints c on the time horizon $\mathscr{T}$, as well as (in-)equality point constraints r_i on a grid of $m+1$ grid points on $\mathscr{T}$,

$$t_0 < t_1 < \ldots < t_{m-1} < t_m := t_\mathrm{f}, \qquad m \in \mathbb{N},\ m \geq 1. \tag{2}$$

The direct multiple shooting method is applied to discretize the control trajectory $u(\cdot)$ to make this infinite dimensional problem computationally accessible.

2.2 Direct Multiple Shooting Discretization

Control Discretization A discretization of the control trajectory $u(\cdot)$ on the *shooting grid* (2) is introduced, using *control parametersq*$_i \in \mathbb{R}^{n_i^\mathrm{q}}$ and *base functions* $b_i : \mathscr{T} \times \mathbb{R}^{n_i^\mathrm{q}} \to \mathbb{R}^{n^\mathrm{u}}$. Examples are piecewise constant or linear functions.

$$u(t) := \textstyle\sum_{j=1}^{n_i^\mathrm{q}} b_{ij}(t,q_{ij}), \qquad t \in [t_i,t_{i+1}] \subseteq \mathscr{T},\ 0 \leq i \leq m-1. \tag{3}$$

State Parameterization In addition to the control parameter vectors, we introduce state vectors $s_i \in \mathbb{R}^{n^\mathrm{x}}$ in all shooting nodes serving as initial values for m IVPs

$$\dot{x}_i(t) = f(t,x_i(t),q_i), \quad x_i(t_i) = s_i \qquad t \in [t_i,t_{i+1}] \subseteq \mathscr{T}, 0 \leq i \leq m-1. \tag{4}$$

This parameterization of the process trajectory $x(\cdot)$ will in general be discontinuous on $\mathscr{T}$. Continuity is ensured by introduction of additional *matching conditions*

$$x_i(t_{i+1};\, t_i,s_i,q_i) - s_{i+1} = 0, \qquad 0 \leq i \leq m-1, \tag{5}$$

where $x_i(t_{i+1};\, t_i,s_i,q_i)$ denotes the evaluation of the i-th state trajectory $x_i(\cdot)$ at time t_{i+1} depending on the start time t_i, initial value s_i, and control parameters q_i.

Constraint Discretization The path constraints of problem (1) are enforced on the nodes of the shooting grid (2) only. It can be observed that in general this formulation already leads to a solution that satisfies the path constraints on the whole of $\mathscr{T}$.

$$0 \leqq r_i(t_i, s_i, q_i), \quad 0 \le i \le m-1, \qquad 0 \leqq r_m(t_m, s_m). \tag{6}$$

Separable Objective The objective function shall be separable with respect to the shooting grid structure,

$$l(x(\cdot), u(\cdot)) = \textstyle\sum_{i=0}^{m} l_i(s_i, q_i). \tag{7}$$

In general, $l(\cdot)$ will be a Mayer type function or a Lagrange type integral function. For both types, a separable formulation is easily found.

Summarizing, the discretized multiple shooting optimal control problem can be cast as a nonlinear problem

$$\begin{aligned} \min_{w} \quad & \textstyle\sum_{i=0}^{m} l_i(w_i) && \text{(8a)} \\ \text{s.t.} \quad & 0 = x_i(t_{i+1};\, t_i, w_i) - s_{i+1} \qquad 0 \le i \le m-1 && \text{(8b)} \\ & 0 \leqq r_i(w_i) \qquad 0 \le i \le m && \text{(8c)} \end{aligned}$$

with the vector of unknowns $w := (s_1, q_1, \ldots, s_{m-1}, q_{m-1}, s_m)$ and subvectors $w_i := (s_i, q_i)$ for $0 \le i \le m-1$, and $w_m := (s_m)$. The evaluation of the matching condition constraint (8b) requires the solution of the initial value problem (4).

2.3 Block Sparse Quadratic Subproblem

For solving the highly structured NLP (8) we employ methods of SQP type, a long–standing and highly effective method for the solution of NLPs that also allow for much flexibility in exploiting the problem's special structure. SQP methods iteratively progress towards a KKT point of the NLP by solving a linearly constrained local quadratic model of the NLP's Lagrangian [11]. For NLP (8) the local quadratic model of the Lagrangian, to be solved in each step of the SQP method, reads

$$\begin{aligned} \min_{\delta w} \quad & \tfrac{1}{2} \textstyle\sum_{i=0}^{m} \delta w_i' B_i \delta w_i + g_i' \delta w_i && \text{(9a)} \\ \text{s.t.} \quad & 0 = X_i(w_i)\delta w_i - \delta s_{i+1} - h_i(w_i), \qquad 0 \le i \le m-1, && \text{(9b)} \\ & 0 \leqq R_i(w_i)\delta w_i - r_i(w_i), \qquad 0 \le i \le m, && \text{(9c)} \end{aligned}$$

with the following notations for vector of unknowns δw and its components

$$\delta w_i := (\delta s_i, \delta q_i),\ 0 \le i \le m-1, \qquad \delta w_m := \delta s_m, \tag{10}$$

reflecting the notation used in (8), and with vectors h_i denoting the residuals

$$h_i(w_i) := x_i(t_{i+1};\, t_i, w_i) - s_{i+1}. \tag{11}$$

The matrices B_i denote the node Hessians or suitable approximations, cf. [3], and the vectors g_i denotes the node gradients of the objective function, while matrices X_i, R_i^{eq}, and R_i^{in} denote linearizations of the constraint functions obtained in w_i,

$$B_i \approx \frac{\mathrm{d}^2 l_i(w_i)}{\mathrm{d}w_i^2}, \quad g_i := \frac{\mathrm{d}l_i(w_i)}{\mathrm{d}w_i}, \quad R_i := \frac{\mathrm{d}r_i(w_i)}{\mathrm{d}w_i}, \quad X_i := \frac{\partial x_i(t_{i+1};\, t_i, w_i)}{\partial w_i}. \tag{12}$$

The computation of the *sensitivity matrices* X_i requires the computation of derivatives of the solution of IVP (4) with respect to the w_i. Consistency of the derivatives is ensured by applying the principle of *internal numerical differentiation* (IND) [1].

3 Block Sparse Quadratic Programming

3.1 Classical Condensing

In the classical *condensing* algorithm [3, 10] that works as a preprocessing step to obtain a small dense QP from the block sparse one, the matching conditions (9b) are used for block Gaussian elimination of the steps of the additionally introduced state variables $(\delta s_1, \dots, \delta s_m)$. The resulting dense QP has $n^{\text{x}} + mn^{\text{q}}$ unknowns instead of $m(n^{\text{x}} + n^{\text{q}})$ ones, is usually densely populated, and suited for solution with any standard QP code such as the null–space active–set codes QPOPT [7], qpOASES [4], or BQPD [5]. As we will see in section 4, for MIOCPs with many controls parameters (i.e. large dimension n^{q}) resulting from the *outer convexification* of integer control functions, the achieved reduction of the QP's size is marginal, however.

3.2 The KKT System's Block Sparse Structure

In this section we present an alternative approach at solving the KKT system of QP (9) found in [16, 17] where it was employed inside an interior–point method. We derive in detail the necessary elimination steps that will ultimately retain the duals of the matching conditions only. In this sense, the approach is *complementary* to the classical *condensing* algorithm. For optimal control problems with dimensions $n^{\text{q}} \geq n^{\text{x}}$, the presented approach obviously is computationally more favorable than retaining unknowns of dimension n^{q}. In contrast to [16, 17] we employ this factorization approach inside an active–set method, and intend to further adapt it to this case by exploitation of simple bounds and derivation of matrix updates in a further publication.

For a given active set, the KKT system of the QP (9) to be solved for the primal step δw_i and the dual step $(\delta\lambda, \delta\mu)$ reads for $0 \leq i \leq m$

$$P_i'\delta\lambda_{i-1} + B_i(-\delta w_i) + R_i'\delta\mu_i + X_i'\delta\lambda_i = B_i w_i + g_i =: \bar{g}_i, \tag{13a}$$

$$R_i(-\delta w_i) = R_i w_i - r_i =: \bar{r}_i, \tag{13b}$$

$$X_i(-\delta w_i) + P_{i+1}(-\delta w_{i+1}) = X_i w_i + P_{i+1} s_{i+1} - h_i =: \bar{h}_i. \tag{13c}$$

with multipliers $\delta\lambda \in \mathbb{R}^{n^x}$ for the matching conditions (9b) and $\delta\mu \in \mathbb{R}^{n_i^r}$ for the active point constraints (9c). The projection matrices P_i are defined as

$$P_i := \begin{pmatrix} -I & 0 \end{pmatrix} \in \mathbb{R}^{n^x \times (n^x + n^q)},\ 1 \leq i \leq m, \tag{14}$$

and as $P_0 := 0 \in \mathbb{R}^{n^x \times (n^x+n^q)}$, $P_{m+1} := 0 \in \mathbb{R}^{n^x \times n^x}$ for the first and last shooting nodes, respectively. In the following, all matrices and vectors are assumed to comprise the components of the active set only. To avoid the need for repeated special treatment of the first and last shooting node throughout this paper, we introduce the following conventions that make equation (13) hold also for the border cases $i = 0$ and $i = m$:

$$\delta\lambda_{-1} := 0 \in \mathbb{R}^{n^x}, \quad \lambda_{-1} := 0 \in \mathbb{R}^{n_x}, \quad \delta\lambda_m := 0 \in \mathbb{R}^{n^x}, \quad \lambda_m := 0 \in \mathbb{R}^{n^x}, \tag{15a}$$

$$\delta w_{m+1} := 0 \in \mathbb{R}^{n^x},\ w_{m+1} := 0 \in \mathbb{R}^{n^x}, \quad h_m := 0 \in \mathbb{R}^{n^x}, \quad X_m := 0 \in \mathbb{R}^{n^x \times n^x}. \tag{15b}$$

3.3 Hessian Projection Schur Complement factorization

Hessian Projection Step Under the assumption that the number of active point constraints does not exceed the number of unknowns, i.e. the active set is not degenerate, we can perform QR factorizations of the point constraints matrices R_i,

$$R_i Q_i = \begin{pmatrix} R_i^{R\prime} & 0 \end{pmatrix}, \qquad Q_i := \begin{pmatrix} Y_i & Z_i \end{pmatrix}. \tag{16}$$

Here Q_i are a unitary matrices and R_i^R is upper triangular. We partition δw_i into its range space part δw_i^Y and its null space part δw_i^Z, where the identity $\delta w_i = Y_i \delta w_i^Y + Z_i \delta w_i^Z$ holds. We find δw_i^Y from the range space projection of (13b)

$$R_i(-\delta w_i) = -R_i^R \delta w_i^Y = \bar{r}_i. \tag{17}$$

We transform the KKT system onto the null space of R_i by substituting $Y_i \delta w_i^Y + Z_i \delta w_i^Z$ for δw_i and solving for δw_i^Z. We find for the matching conditions (13c)

$$-X_i Z_i \delta w_i^Z - P_{i+1} Z_i \delta w_{i+1}^Z = \bar{h}_i + X_i Y_i \delta w_i^Y + P_{i+1} Y_i \delta w_{i+1}^Y \tag{18}$$

to be solved for δw_i^Z once δw_{i+1}^Z is known. For stationarity (13a) we find

$$Z_i' P_i' \delta\lambda_{i-1} - Z_i' B_i Z_i \delta w_i^Z + Z_i' R_i' \mu_i + Z_i' X_i' \delta\lambda_i = Z_i' \bar{g}_i + Z_i' B_i Y_i \delta w_i^Y \tag{19}$$

$$\text{and} \qquad Y_i' R_i' \delta\mu_i = -Y_i' (B_i \delta w_i + P_i' \delta\lambda_{i-1} - X_i' \delta\lambda_i + \bar{g}_i). \tag{20}$$

Therein, $Z_i' R_i = 0$ and $Y_i' R_i' = R_i^R$ Thus (19) can be solved for $\delta\lambda_i$ once δw_i and $\delta\lambda_{i-1}$ are known, while (20) can be used to determine the point constraints multipliers $\delta\mu_i$. Let thus null space projections be defined as follows:

$$\tilde{B}_i := Z_i' B_i Z_i, \qquad \tilde{g}_i := Z_i' \bar{g}_i + Z_i' B_i Y_i \delta w_i^Y, \qquad 0 \leq i \leq m, \tag{21a}$$

$$\tilde{X}_i := X_i Z_i, \qquad \tilde{h}_i := \bar{h}_i + X_i Y_i \delta w_i^Y + P_{i+1} Y_i \delta w_{i+1}^Y, \qquad 0 \leq i \leq m-1, \tag{21b}$$

$$\tilde{P}_i := P_i Z_i, \qquad 0 \leq i \leq m-1. \tag{21c}$$

With this notation the projection of the KKT system on the null space of the point constraints can be read from equations (18) and (19) for $0 \leq i \leq m-1$ as

$$\tilde{P}_i' \delta\lambda_{i-1} + \tilde{B}_i(-\delta w_i^Z) + \tilde{X}_i' \delta\lambda_i = \tilde{g}_i, \tag{22a}$$

$$\tilde{X}_i(-\delta w_i^Z) + \tilde{P}_{i+1}(-\delta w_{i+1}^Z) = \tilde{h}_i. \tag{22b}$$

Schur Complement Step In (22a) the elimination of δw^Z is possible using a Schur complement step, provided that the reduced Hessians $\tilde{B}_i$ are positive definite. We find

$$(-\delta w_i^Z) = \tilde{B}_i^{-1}(\tilde{g}_i - \tilde{P}_i' \delta\lambda_{i-1} - \tilde{X}_i' \delta\lambda_i) \tag{23}$$

depending on the knowledge of $\delta\lambda_i$. Inserting into (22b) and collecting for $\delta\lambda_i$ yields

$$\begin{aligned} &\tilde{X}_i\tilde{B}_i^{-1}\tilde{P}_i'\delta\lambda_{i-1} + (\tilde{X}_i\tilde{B}_i^{-1}\tilde{X}_i' + \tilde{P}_{i+1}\tilde{B}_{i+1}^{-1}\tilde{P}_{i+1}')\delta\lambda_i + \tilde{P}_{i+1}\tilde{B}_{i+1}^{-1}\tilde{X}_{i+1}'\delta\lambda_{i+1} \\ =& -\tilde{h}_i + \tilde{X}_i\tilde{B}_i^{-1}\tilde{g}_i + \tilde{P}_{i+1}\tilde{B}_{i+1}^{-1}\tilde{g}_{i+1} \end{aligned} \tag{24}$$

With Cholesky factorizations $\tilde{B}_i = R_i^{\mathrm{B}\,\prime} R_i^{\mathrm{B}}$ we define the following symbols

$$\begin{aligned} \hat{X}_i &:= \tilde{X}_i R_i^{\mathrm{B}\,-1}, & A_i &:= \tilde{X}_i\tilde{B}_i^{-1}\tilde{X}_i' + \tilde{P}_{i+1}\tilde{B}_{i+1}^{-1}\tilde{P}_{i+1}' &&= \hat{X}_i\hat{X}_i' + \hat{P}_{i+1}\hat{P}_{i+1}', \\ \hat{P}_i &:= \tilde{P}_i R_i^{\mathrm{B}\,-1}, & B_i &:= \tilde{X}_i\tilde{B}_i^{-1}\tilde{P}_i' &&= \hat{X}_i\hat{P}_i', \\ \hat{g} &:= R_i^{\mathrm{B}\,-T}\tilde{g}_i, & a_i &:= -\tilde{h}_i + \tilde{X}_i\tilde{B}_i^{-1}\tilde{g}_i + \tilde{P}_{i+1}\tilde{B}_{i+1}^{-1}\tilde{g}_{i+1} &&= -\tilde{h}_i + \hat{X}_i\hat{g}_i + \hat{P}_{i+1}\hat{g}_{i+1}. \end{aligned} \tag{25}$$

Equation (24) may then be written in terms of these values for $0 \le i \le m-1$ as

$$B_i\delta\lambda_{i-1} + A_i\delta\lambda_i + B_{i+1}'\delta\lambda_{i+1} = a_i. \tag{26}$$

Solving the Block Tridiagonal System In the symmetric positive definite banded system (26), only the matching condition duals $\delta\lambda_i \in \mathbb{R}^{n^{\mathrm{x}}}$ remain as unknowns. In classical condensing, exactly these matching conditions were used for elimination of a part of the primal unknowns. System (26) can be solved for $\delta\lambda$ by means of a block tridiagonal Cholesky factorization and two backsolves.

Recovering the Block Sparse QP's Solution Once $\delta\lambda$ is known, the step δw^Z can be recovered using equation (23). The full primal step δw is then obtained from $\delta w = Y\delta w^Y + Z\delta w^Z$. The constraint multipliers step $\delta\mu$ is recovered using (20).

3.4 Computational Complexity

In the left part of table 1 a detailed list of the linear algebra operations required to carry out the individual steps of the complementary condensing method can be found. The number of floating point operations (FLOPs) required per shooting node, depending on the system's dimensions $n = n^{\mathrm{x}} + n^{\mathrm{q}}$ and n_i^{r}, is given in the right part of table 1. The numbers n^{y} and n^{z} with $n^{\mathrm{y}} + n^{\mathrm{z}} = n_i^{\mathrm{r}}$ denote the range–space and null-space dimension in (16), respectively. The proposed method's runtime complexity is $O(m)$, in sharp contrast to the classical condensing method's $O(m^2)$, as the shooting grid length m does not appear explicitly in table 1.

4 Example: A Vehicle Mixed–Integer Optimal Control Problem

In this section we formulate a vehicle control problem as a test bed for the presented approach to solving the block sparse QPs.

Table 1. Left: Number of factorizations (dc), backsolves (bs), multiplications (*), and additions (+) required per shooting node. **Right:** Number of FLOPs required per shooting node.

	Matrix				Vector		
Action	dc	bs	*	+	bs	*	+
Decompose R_i	1	–	–	–			
Solve for δw^Y, $Y\delta w^Y$					1	1	–
Build $\tilde{B}_i$	–	–	2	–			
Build $\tilde{X}_i$, $\tilde{P}_i$	–	–	2	–			
Build $\tilde{g}_i$, $\tilde{h}_i$					–	4	3
Decompose $\tilde{B}_i$	1	–	–	–			
Build $\hat{X}_i$, $\hat{P}_i$	–	1	1	–			
Build A_i, B_i	–	–	3	1			
Build $\hat{g}_i$, a_i					–	3	2
Decompose (26)	1	–	–	–			
Solve for $\delta\lambda_i$					2	–	–
Solve for δw_i^Z, $Z\delta w_i^Z$					2	3	2
Solve for $\delta\mu_i$					1	4	3

Action	Floating point operations
Decompose R_i	$n_i^{\mathrm{r}2}(n - \frac{1}{3}n_i^{\mathrm{r}})$
Solve for $Y\delta w^Y$	$n_i^{\mathrm{r}} n^{\mathrm{y}} + n^{\mathrm{y}} n$
Build $\tilde{B}_i$	$n^{\mathrm{z}2} n + n^{\mathrm{z}} n^2$
Build $\tilde{X}_i$, $\tilde{P}_i$	$2n^{\mathrm{x}} n^{\mathrm{z}} n$
Build $\tilde{g}_i$, $\tilde{h}_i$	$2n^{\mathrm{x}} n + n^{\mathrm{z}} n + n^2 + 2n^{\mathrm{x}} + n$
Decompose $\tilde{B}_i$	$\frac{1}{3}n^{\mathrm{z}3}$
Build $\hat{X}_i$, $\hat{P}_i$	$2n^{\mathrm{x}} n^{\mathrm{z}2}$
Build A_i, B_i	$3n^{\mathrm{x}2} n^{\mathrm{z}} + n^{\mathrm{x}2}$
Build $\hat{g}_i$, a_i	$n^{\mathrm{z}2} + 2n^{\mathrm{x}} n^{\mathrm{z}} + 2n^{\mathrm{x}}$
Decompose (26)	$\frac{4}{3}n^{\mathrm{x}3}$
Solve for $\delta\lambda_i$	$2 \cdot 2n^{\mathrm{x}2}$
Solve for $Z\delta w_i^Z$	$n^{\mathrm{z}2} + 2n^{\mathrm{x}} n^{\mathrm{z}} + n^{\mathrm{z}} n + 2n^{\mathrm{z}}$
Solve for $\delta\mu_i$	$n^{\mathrm{r}}{}_i n^{\mathrm{y}} + 2n^{\mathrm{x}} n + n^{\mathrm{y}} n + n^2 + 3n$

Exemplary Vehicle Mixed–Integer Optimal Control Problem We consider a simple dynamic model of a car driving with velocity v on a straight lane with varying slope γ. The optimizer excerts control over the engine and brake torque rates of change R_{eng} and R_{brk}, and the gear choice y. The state dimension is $n^{\mathrm{x}} = 3$, and we consider different numbers of available gears to scale the problems control dimension $n^{\mathrm{q}} \geq 3$.

$$\dot{v}(t) = \frac{1}{m}\left(\frac{i_{\mathrm{A}}}{r}\left(i_{\mathrm{T}}(y)\eta_{\mathrm{T}}(y)M_{\mathrm{eng}} - M_{\mathrm{brk}} - i_{\mathrm{T}}(y)M_{\mathrm{fric}}\right) - M_{\mathrm{air}} - M_{\mathrm{road}}\right) \tag{27a}$$

$$\dot{M}_{\mathrm{eng}}(t) = R_{\mathrm{acc}}(t), \qquad \dot{M}_{\mathrm{brk}}(t) = R_{\mathrm{brk}}(t) \tag{27b}$$

Herein m is the vehicle's mass, i_{A} and $i_{\mathrm{T}}(y)$ are the rear axle and gearbox transmission ratios. The amount of engine friction is denoted by M_{fric}, a nonlinear function of the engine speed. By $M_{\mathrm{air}} := \frac{1}{2}c_{\mathrm{w}}A\rho_{\mathrm{air}}v^2(t)$ we denote air resistance, c_{w} being the aerodynamic shape coefficient, A the effective flow surface, and ρ_{air} the air density. Finally $M_{\mathrm{road}} = mg(\sin\gamma(t) + f_{\mathrm{r}}\cos\gamma(t))$ accounts for downhill force and tyre friction, g being the gravity constant and f_{r} the coefficient of rolling friction.

On a predefined track with varying slope, we minimize a weighted sum of travel time and fuel consumption, subject to velocity and engine speed constraints making the gear choice nontrivial.

Run Time Complexity Clearly from table 2 it can be seen that the classical condensing algorithm will be suitable for problems with limited grid lengths m and with considerably less controls than states, i.e. $n^{\mathrm{q}} \ll n^{\mathrm{x}}$, which is exactly contrary to the situation encountered when applying outer convexification to MIOCPs. Nonetheless, using this approach we could solve several challenging mixed–integer optimal control problems to optimality with little computational effort, as reported in [9, 12, 14].

Table 2. Run time complexity of classical condensing and a dense active–set QP solver.

Action	Run time complexity
Computing the Hessian $\overline{\overline{B}}$	$O(m^2n^3)$
Computing the Constraints $\overline{X}$, $\overline{R}$	$O(m^2n^3)$
Dense QP solver, startup	$O((mn^{\text{q}}+n^{\text{x}})^3)$
Dense QP solver, per iteration	$O((mn^{\text{q}}+n^{\text{x}})^2)$
Recovering δv	$O(mn^{\text{x}2})$

Sparsity In table 3 the dimensions and amount of sparsity present in the Hessian and constraints matrices are given for the exemplary problem for 6 and 16 available gears. A grid length of $m=20$ was used. As can be seen in the left part, the QP (9) is only sparsely populated for this example problem, with the number of nonzero elements (nnz) never exceeding 3 percent. After classical condensing, sparsity has been lost as expected. Had the overall dimension of the QP reduced considerably, as is the case for optimal control problems with $n^{\text{x}} \gg n^{\text{q}}$, that would be of no concern. For our MIOCP with outer convexification, however, the results shown in tables 2 and 3 indicate a considerable run time increase for larger m or n^{q} is to be expected.

Table 3. Dimensions and number of nonzero elements (nnz) of the block structured QP (9) and the condensed QP for the exemplary vehicle control problem. Here $m=20$, $n^{\text{x}}=3$.

		Block sparse		Condensed		Dense QP solver
n^{q}	Matrix	Size	nnz	Size	nnz	nnz seen
2+6	Hessian	223×223	$1,419$	163×163	$13,366$	$13,366$ (27%)
	Constraints	438×223	$1,535$	378×163	$13,591$	$61,614$ (63%)
2+16	Hessian	423×423	$6,623$	363×363	$66,066$	$66,066$ (37%)
	Constraints	858×423	$3,731$	798×363	$131,769$	$289,674$ (80%)

Implementation Run Times The classical condensing algorithm as well as the QP solver QPOPT [7] are implemented in ANSI C and translated using gcc 4.3.3 with optimization level `-O3`. The linear algebra package ATLAS was used for BLAS operations. The proposed complementary condensing algorithm was preliminarily implemented in MATLAB© (Release 2008b). All run times have been obtained on a Pentium 4 machine at 3 GHz under SuSE Linux 10.3. The resulting run times shown in table 4 support our conclusions drawn from table 3. For $m=30$ as well as for $m=20$ and $n^{\text{q}} \geq 14$ the MATLAB code of our proposed methods beats an optimized C implementation of classical condensing plus QPOPT. In addition, we could solve four instances with $m=30$ or $n^{\text{q}}=18$ that could not be solved before due to active set cycling of the QPOPT solver.

Table 4. Average run time per iteration of the QP solver QPOPT on the condensed QPs (**left**, condensing run times *excluded*), and of a preliminary MATLAB code running proposed method on the block sparse QPs (**right**). "–" indicates cycling of the active set.

	m		
n^{q}	10	20	30
$2+6$	6 ms	31 ms	103 ms
$2+8$	11 ms	58 ms	467 ms
$2+12$	18 ms	226 ms	–
$2+16$	–	–	–

	m		
n^{q}	10	20	30
$2+6$	40 ms	65 ms	100 ms
$2+8$	42 ms	75 ms	110 ms
$2+12$	50 ms	95 ms	140 ms
$2+16$	60 ms	115 ms	170 ms

5 Summary and Future Work

Summarizing the results presented in tables 3 and 4, we have seen that for OCPs with larger dimension n^{q}, the classical $O(m^2n^3)$ condensing algorithm is unable to significantly reduce the QPs size. Worse yet, the condensed QP is densely populated. As a consequence, the dense QP solver's performance, exemplarily tested using QPOPT, is worse than what can be achieved by a suitable exploitation of the sparse block structure for the case $n^{\mathrm{q}} \geq n^{\mathrm{x}}$.

We presented an alternative $O(mn^3)$ factorization of the block sparse KKT system due to [16, 17], named *complementary condensing* in the context of MIOCPs. By theoretical analysis as well as by preliminary implementation we provided evidence that the proposed approach is able to challenge the run times of the classical condensing algorithm.

The complementary condensing approach for solving the QP's KKT system is embedded in an active set loop. In our preliminary implementation, a new factorization of the KKT system is computed in $O(mn^3)$ time in every iteration of the active set loop. Nonetheless, the achieved computation times are attractive for larger values of m or n^{q}. To improve the efficiency of this active set method further, several issues have to be addressed. Exploiting *simple bounds* on the unknowns will reduce the size of the matrices B_i, R_i, and X_i involved. For dense null–space and range–space methods it is common knowledge that certain factorizations can be updated after an active set change in $O(n^2)$ time. Such techniques would essentially relieve the active–set loop from all matrix-only operations, yielding $O(mn^2)$ active set iterations with only an initial factorization in $O(mn^3)$ time necessary. A forthcoming publication shall investigate into this topic.

References

1. J. ALBERSMEYER AND H. BOCK, *Sensitivity Generation in an Adaptive BDF-Method*, in Modeling, Simulation and Optimization of Complex Processes: Proc. 3rd Int. Conf. on High Performance Scientific Computing, Hanoi, Vietnam, 2008, pp. 15–24.
2. L. BIEGLER, *Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation*, Comp. Chem. Eng., 8 (1984), pp. 243–248.

3. H. Bock and K. Plitt, *A Multiple Shooting algorithm for direct solution of optimal control problems*, in Proc. 9th IFAC World Congress Budapest, 1984, pp. 243–247.
4. H. Ferreau, H. Bock, and M. Diehl, *An online active set strategy for fast parametric quadratic programming in MPC applications*, in Proc. IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems, Grenoble, 2006.
5. R. Fletcher, *Resolving degeneracy in quadratic programming*, Numerical Analysis Report NA/135, University of Dundee, Dundee, Scotland, 1991.
6. M. Gerdts, *A variable time transformation method for mixed-integer optimal control problems*, Optimal Control Applications and Methods, 27 (2006), pp. 169–182.
7. P. Gill, W. Murray, and M. Saunders, *User's Guide For QPOPT 1.0: A Fortran Package For Quadratic Programming*, 1995.
8. E. Hellström, M. Ivarsson, J. Åslund, and L. Nielsen, *Look-ahead control for heavy trucks to minimize trip time and fuel consumption*, Control Eng. Pract., 17 (2009), pp. 245–254.
9. C. Kirches, S. Sager, H. Bock, and J. Schlöder, *Time-optimal control of automobile test drives with gear shifts*, Opt. Contr. Appl. Meth. (2010). DOI 10.1002/oca.892.
10. D. Leineweber, I. Bauer, H. Bock, and J. Schlöder, *An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part I: Theoretical aspects*, Computers and Chemical Engineering, 27 (2003), pp. 157–166.
11. J. Nocedal and S. Wright, *Numerical Optimization*, Springer, 2nd ed., 2006.
12. S. Sager, *Numerical methods for mixed–integer optimal control problems*, Der andere Verlag, Tönning, Lübeck, Marburg, 2005.
13. S. Sager, *Reformulations and algorithms for the optimization of switching decisions in nonlinear optimal control*, Journal of Process Control, 19 (2009), pp. 1238–1247.
14. S. Sager, C. Kirches, and H. Bock, *Fast solution of periodic optimal control problems in automobile test-driving with gear shifts*, in Proc. 47th IEEE CDC, Cancun, Mexico, 2008, pp. 1563–1568.
15. S. Sager, G. Reinelt, and H. Bock, *Direct methods with maximal lower bound for mixed-integer optimal control problems*, Math. Prog., 118 (2009), pp. 109–149.
16. M. Steinbach, *Fast recursive SQP methods for large-scale optimal control problems*, PhD thesis, Universität Heidelberg, 1995.
17. ———, *Structured interior point SQP methods in optimal control*, Zeitschrift für Angewandte Mathematik und Mechanik, 76 (1996), pp. 59–62.
18. ———, *Tree-sparse convex programs*, Math. Methods Oper. Res., 56 (2002), pp. 347–376.
19. S. Terwen, M. Back, and V. Krebs, *Predictive powertrain control for heavy duty trucks*, in Proc. IFAC Symposium in Advances in Automotive Control, Salerno, Italy, 2004, pp. 451–457.
20. J. Till, S. Engell, S. Panek, and O. Stursberg, *Applied hybrid system optimization: An empirical investigation of complexity*, Control Eng. Pract., 12 (2004), pp. 1291–1303.