# A Structured Interior Point SQP Method for Nonlinear Optimal Control Problems*

Marc C. Steinbach†

**Abstract**

Direct boundary value problem methods in connection with SQP iteration have proven very successful in solving nonlinear optimal control problems. Such methods use parameterized control functions, discretize the state differential equations by, e.g., multiple shooting or collocation, and treat the discretized BVP as an equality-constraint in a large, nonlinear, constrained optimization problem. In realistic applications several thousands of variables can appear in the NLP. Solution by standard techniques is therefore impractical. A careful choice of the discretization leads to QP subproblems possessing a very special $m$-stage block-sparse structure, where $m$ is the grid size. The paper presents a recursive solution algorithm that fully exploits this QP sparseness to generate a factorization of the inverse of the KKT matrix in $\mathcal{O}(m)$ operations. A structure-preserving primal-dual barrier method is proposed for treating the generally large number of state and control inequality constraints.

# Introduction

Consider the Mayer control problem with multi-point boundary conditions and path constraints (usually including state and control box constraints):

$$
\begin{aligned}
\phi(x(T)) &= \min, && (1)\\
\dot{x}(t) - f(x(t), u(t)) &= 0, && (2)\\
r(x(t_1), \ldots, x(t_k)) &= 0, && (3)\\
g(x(t), u(t)) &\geq 0. && (4)
\end{aligned}
$$

As described below, we will discretize the problem by direct multiple shooting or collocation schemes and solve the resulting nonlinear optimization problem by an

*To appear in: Computational Optimal Control, R. Bulirsch and D. Kraft, eds., International Series of Numerical Mathematics, pp. 213–222, Birkhäuser publishers, Basel, Boston, Berlin, 1994.

†Academic Assistant. Interdisciplinary Center for Scientific Computation, University of Heidelberg, Im Neuenheimer Feld 368, 69120 Heidelberg, Germany. The research was supported in part by the Deutsche Forschungsgemeinschaft (DFG).

SQP iteration. This numerical approach has proven very effective during the last decade, and several methods based on it have been developed [2, 8, 1, 16].

Applying the Karush-Kuhn-Tucker necessary conditions to the QP subproblems yields very large, sparse KKT systems with many inequality constraints. More specific, the numbers of variables and of nonzero matrix elements both grow linearly with the number $m$ of discretization stages. Therefore QP solution dominates the computations by far when standard $\mathcal{O}(m^3)$ algorithms are applied, and exploitation of the sparseness is crucial for an efficient SQP method.

The first structured SQP method for optimal control problems was developed in [14, 2] and implemented in the direct multiple shooting code MUSCOD. Here a generalized condensing algorithm eliminates all but the initial state variables from the QP in a recursive projection. An active set QP solver then operates on the condensed system. MUSCOD has been successfully applied, e.g., to solve moderate size robot trajectory optimization problems [10, 11, 15]. A different sparse SQP approach is reported in [1] for direct collocation. The authors apply a multifrontal algorithm to obtain a sparse KKT matrix factorization, which is then re-used in a Schur complement method to account for active set changes.

Here we consider a specific $m$-stage block-sparse structure that can be fully exploited by the SQP method. This rather natural structure is obtained under mild restrictions on the discretization whenever the boundary conditions satisfy a second order *separability condition*. In this paper we restrict ourselves to the case of first order decoupling, $r(x_1, \ldots, x_k) = (r_1(x_1), \ldots, r_k(x_k))$, which covers an important class of application problems. The more general case of second order decoupling, $r(x_1, \ldots, x_k) = r_1(x_1) + \cdots + r_k(x_k)$, includes almost all practical nonlinear optimal control problems (in particular, periodic ones), and is already implemented in our recursive solver. However, this case exhibits basic differences in theory and practice, and will be treated in a subsequent paper. (See also the remarks after theorem 1 and after the description of the solver.)

# The direct BVP approach

The *infinite* optimal control problem is converted to a *finite* nonlinear optimization problem (NLP) by parameterizing the controls and discretizing the states.

1) Choose a grid $0 = \tau_1 < \cdots < \tau_m = T$ of $m \geq k$ points which include all instances where boundary conditions are evaluated: $t_i = \tau_{j(i)}$ for some $j(i)$.

2) Choose a finite-dimensional space of admissible controls, defined by fixed *local* base functions $v_j$ and free control parameters $u_j$:

$$u(t) = v_j(t; u_j) \quad \text{on } I_j = [\tau_j, \tau_{j+1}). \tag{5}$$

3) Define a discrete trajectory $x_j = x(\tau_j)$ and *local* approximations $y_j$ to the ODE solution. Here $y_j(\tau_j; x_j, z_j) = x_j$, and the additional *collocation variables* $z_j$ satisfy *collocation conditions* $c_j(x_j, z_j, u_j) = 0$. Couple the stages by matching

conditions

$$h_j(x_j, z_j, x_{j+1}) = y_j(\tau_{j+1}; x_j, z_j) - x_{j+1} = 0. \qquad (6)$$

In the multiple shooting case $y_j$ solves the IVP $y_j(\tau_j; x_j, u_j) = x_j$, $\dot{y}_j = f(y_j, v_j)$ on $I_j$, and $u_j$ replaces $z_j$ in $y_j, h_j$.

4) Discretize the remaining problem functions by substituting $u_j, v_j, x_j, z_j$ into the definitions, and possibly include collocation conditions: [1]

$$\min \phi(x_m), \quad e_j(x_j, z_j, u_j) = \begin{pmatrix} r_j(x_j) \\ c_j(x_j, z_j, u_j) \end{pmatrix}, \quad g_j(x_j, u_j) = g(x_j, v_j(\tau_j; u_j)). \qquad (7)$$

Replacing continuous path constraints by pointwise constraints may lead to small violations between grid points. This can affect the SQP convergence order depending on the problem, parameterization, and grid choice. Note that the discretization scheme must not include any stage-coupling terms, except for $x_{j+1}$ in $h_j$ satisfing $\partial^2 h_j / \partial(x_j, u_j) \partial x_{j+1} = 0$. Precisely this restriction yields our special structure; it is usually not satisfied for collocation schemes applied by other authors. Details of our collocation approach will be given in a forthcoming paper.

# SQP method for the structured NLP

Define $F_1 = \phi$, $F_2 = (e_1, h_1, \ldots, e_{m-1}, h_{m-1}, e_m)$, $F_3 = (g_1, \ldots, g_m)$. A numerical solution $y = (x_1, z_1, u_1, \ldots, x_m, z_m, u_m)$ of the constrained NLP

$$\min F_1(y) \qquad \text{subject to} \qquad F_2(y) = 0, \quad F_3(y) \geq 0, \qquad (8)$$

is computed by applying a damped SQP iteration, $y^{k+1} = y^k + t^k \Delta y^k$, where $\Delta y^k$ solves an approximating quadratic subproblem with linear constraints:

$$\min_{\Delta y} \frac{1}{2} \Delta y^T H^k \Delta y + J_1^k \Delta y \qquad \text{s.t.} \qquad \begin{cases} J_2^k \Delta y + F_2^k = 0, \\ J_3^k \Delta y + F_3^k \geq 0. \end{cases} \qquad (9)$$

Here $F_i^k = F_i(y^k)$, $J_i^k = F_i'(y^k)$, and $H^k \approx L_{yy}(y^k, w^k)$ approximates the Hessian of the Lagrangian. The following theorem establishes our special QP structure.

**Theorem 1** *The jacobians $J_2^k, J_3^k$ and Hessian $L_{yy}^k$ are* block-diagonal*, with the only exception of stage-coupling superdiagonal blocks $\partial h_j / \partial x_{j+1} = -I$ in $J_2^k$.*

*Proof.* This follows from the definitions of functions $F_i$ and constituents $\phi, h_j, g_j, e_j$. Except for $h_j$ they involve only local variables $x_j, z_j, u_j$. The Hessian $L_{yy}^k$ is a linear combination of Hessians of $\phi, h_j, g_j, e_j$. ∎

*Remark:* In the case of second order decoupled boundary conditions we redefine $F_2 = (c_1, h_1 \ldots, c_{m-1}, h_{m-1}, r)$ and get a full row of blocks in $J_2^k$ representing globally coupled constraints. All other structures are preserved.

[1] For notational simplicity assume $\tau_j = t_j$ and introduce zero-dimensional variables $z_m, u_m$.

From now on we assume that collocation variables and conditions have been eliminated from the discrete QP. After appropriate partitioning of the jacobian and Hessian blocks the QP then fits into the general $m$-stage block-sparse structure (${}^tJ = J^T = \text{transpose}(J)$):

$$\sum_{j=1}^{m}\left[\frac{1}{2}\begin{pmatrix}\Delta x_j\\ \Delta u_j\end{pmatrix}^T\begin{pmatrix}H_j & {}^tJ_j\\ J_j & K_j\end{pmatrix}\begin{pmatrix}\Delta x_j\\ \Delta u_j\end{pmatrix}+\begin{pmatrix}f_j\\ d_j\end{pmatrix}^T\begin{pmatrix}\Delta x_j\\ \Delta u_j\end{pmatrix}\right] = \min_{\Delta x,\Delta u}, \tag{10}$$

$$\begin{pmatrix}G_j & E_j\end{pmatrix}\begin{pmatrix}\Delta x_j\\ \Delta u_j\end{pmatrix} + h_j - \Delta x_{j+1} = 0, \tag{11}$$

$$F_j^x \Delta x_j + e_j^x = 0 \text{ or } \geq 0, \tag{12}$$

$$D_j^u \Delta u_j + e_j^u = 0 \text{ or } \geq 0, \tag{13}$$

$$\begin{pmatrix}F_j^c & D_j^c\end{pmatrix}\begin{pmatrix}\Delta x_j\\ \Delta u_j\end{pmatrix} + e_j^c = 0 \text{ or } \geq 0. \tag{14}$$

In addition to the minimization and matching conditions we distinguish three categories of (local) constraints: separated state constraints (12), separated control constraints (13), coupled constraints on state and control (14). This is necessary for full efficiency of the QP solver; see next section.

Exploitation of the structure is also possible in the nonlinear algorithmic parts. Generation and storage of function values, gradient and Hessian matrices is *completely parallel* for the SQP and interior point iterations, since all computations are local to the individual discretization stages. When the exact Hessian is not available or indefiniteness must be avoided, block-wise rank-2 BFGS updates yield a rank-$2m$ Hessian update. This leads to superlinear SQP convergence with an essentially mesh-independent convergence order. To increase efficiency, the accuracy of interior point QP solution should be adapted to the SQP needs, if possible also the accuracy of function and gradient evaluation. For details see [14].

# Recursive $\mathcal{O}(m)$ solution of block-sparse QP

In this section we treat the purely equality-constrained QP. Assume for simplicity that a unique minimum exists, i.e., $J_2^k$ has full rank and $H^k$ is positive definite on $\ker(J_2^k)$. Standard solution techniques for the associated symmetric, indefinite KKT system

$$\Omega z + a = \begin{pmatrix}H^k & {}^tJ_2^k\\ J_2^k & \end{pmatrix}\begin{pmatrix}\Delta y\\ -w_2\end{pmatrix} + \begin{pmatrix}{}^tJ_1^k\\ F_2^k\end{pmatrix} = 0 \tag{15}$$

include the null space or *Schur complement* (SC) method and range space or *projected Hessian* (PH) method. A structured SC method would have to factorize $H^k$ and resulting block-diagonal and block-tridiagonal matrices. Partial projection (on the boundary conditions) would generate a similar but smaller block-tridiagonal matrix. Both techniques require positive-definiteness of $H^k$ on a larger space than

$\ker(J_2^k)$, and, more severely, both destroy the block-sparseness when boundary conditions are only second-order decoupled.

Our solution algorithm combines PH and SC methods in an alternating fashion to obtain a solution of the $m$-stage KKT system (15). In a backward recursion, $k = m(-1)1$, a sequence of similar $k$-stage KKT systems

$$\Omega_k z_k + a_k = 0 \tag{16}$$

is generated.[2] A recursion step "cuts off the last stage" by applying a local projection and computing the Schur complement of the projected Hessian. This generates a block-sparse factorization

$$\Omega^{-1} = \Lambda^T \Pi \Lambda, \qquad \Lambda = \Lambda_1 \ldots \Lambda_m, \quad \Pi = \Pi_1 \ldots \Pi_m. \tag{17}$$

(without producing any fill-in) in the factorization phase, and a transformed right hand side $a_1 = \Lambda a$ in an independent transformation phase. Finally, a forward recursion generates $z = \Lambda^T z_1$ from $z_1 = \Pi a_1$ in the solution phase. Each phase requires $\mathcal{O}(m)$ operations.

The whole procedure is based on the optimal control nature of the KKT system. The initial state and all controls are "free" variables, which determine all following states via the matching conditions. Hence, satisfaction of any constraint must be "controlled" either by local free variables or by free variables in *previous* stages (via the matching conditions). Higher order state constraints, for instance, always produce such a backward effect. When free variables in the *last* stage are not fixed by local constraints, they can be determined by *local* minimization according to the basic concept of optimality of sub-arcs. Our algorithm takes these facts into account when eliminating each variable as early as possible in the backward recursion.

Each recursion step involves six transformation steps. Each transformation reduces the local system size by direct or formal elimination of some variables. Transformations 1 and 2 are independent w.r.t. the stages. They eliminate the local separated constraints and may be performed as parallel pre- (post-)processing before backward (after forward) recursion. Transformation steps 3–6 constitute the genuinely recursive part of the algorithm, where steps 3 and 4 eliminate local coupled constraints, and steps 5 and 6 eliminate matching and control minimization conditions respectively. Steps 3 and 4 of stage $j-1$ can run parallel to steps 5 and 6 of stage $j$.

All projection steps except 5 involve the following operations in the backward recursion. A constraint matrix is factorized, the system is accordingly transformed and partitioned. Some variables are either immediately determined or formally eliminated using the constraint equation. Corresponding multipliers are always formally eliminated. In the forward recursion formally eliminated variables are actually computed and partitions are re-transformed to obtain the original variables. Orthogonal ($LQ$ type) and Cholesky ($LL^T$) factorizations are used in projection and Schur complement steps respectively.

[2] Strictly speaking, all systems are $m$-stage systems, with the last $m-k$ stages remaining unchanged in steps $j \le k$.

## The recursion—step by step

Let us introduce the formal definitions $\lambda_0 := 0$, $(G_m\ E_m) := 0$, $\lambda_m := 0$, $h_m := 0$, and $\Delta x_{m+1} := 0$, with appropriate dimensions. When stage coupling terms are included in the right hand side, the local original KKT system at each stage $j$ reads

$$\begin{bmatrix} H_j & {}^tJ_j & {}^tF_j^x & & {}^tF_j^c & {}^tG_j \\ J_j & K_j & & {}^tD_j^u & {}^tD_j^c & {}^tE_j \\ F_j^x & & & & & \\ & D_j^u & & & & \\ F_j^c & D_j^c & & & & \\ G_j & E_j & & & & \end{bmatrix} \begin{bmatrix} \Delta x_j \\ \Delta u_j \\ \mu_j^x \\ \mu_j^u \\ \mu_j^c \\ \lambda_j \end{bmatrix} + \begin{bmatrix} f_j - \lambda_{j-1} \\ d_j \\ e_j^x \\ e_j^u \\ e_j^c \\ h_j - \Delta x_{j+1} \end{bmatrix} = 0. \tag{18}$$

**Steps 1 and 2.** *Project onto* $\ker(D_j^u)$ *and* $\ker(F_j^x)$ *of local separated control and state constraints.*
Factorize $D_j^u, F_j^x$, transform and partition $\Delta u_j \to (\Delta u_j^1, \Delta u_j^*)$, $\Delta x_j \to (\Delta x_j^1, \Delta x_j^*)$. Determine $\Delta u_j^1, \Delta x_j^1$ immediately, eliminate $\mu_j^u, \mu_j^x$ formally. Eliminating $\Delta x_{j+1}^1$ results in *splitting* of the matching conditions $h_j$: rows corresponding to $\Delta x_{j+1}^1$ are converted to local coupled constraints, and $\lambda_j \to (\lambda_j^1, \lambda_j^*)$ is partitioned accordingly. These operations and recursion in higher stages produce the local KKT system

$$\begin{bmatrix} H_j^{**} & {}^tJ_j^{**} & {}^tC_j & {}^tG_j^3 \\ J_j^{**} & K_j^{**} & {}^tA_j & {}^tE_j^3 \\ C_j & A_j & & \\ G_j^3 & E_j^3 & & \end{bmatrix} \begin{bmatrix} \Delta x_j^* \\ \Delta u_j^* \\ \nu_j \\ \lambda_j^3 \end{bmatrix} + \begin{bmatrix} f_j^{**} - \lambda_{j-1}^* \\ d_j^{**} \\ c_j \\ h_j^3 - \Delta x_{j+1}^3 \end{bmatrix} = 0, \tag{19}$$

$$S_{j+1}\Delta x_{j+1}^3 + [\alpha_{j+1} - \lambda_j^3] = 0. \tag{20}$$

Here the third row combines original local coupled constraints with such resulting from splitting in steps 2 and 4 of stage $j+1$, and the additional equation remains from stage $j+1$ after the final step 6. The whole recursion is started in stage $m$ with a zero-dimensional augmentation $S_{m+1} := 0$, $\alpha_{m+1} := 0$.

**Steps 3 and 4.** *Project onto* $\ker(A_j)$ *of control part and onto* $\ker(\bar{C}_j^x)$ *of resulting state part of projected local coupled constraints.*
Some care must be taken in this part of the algorithm. We wish to eliminate as many control variables as possible. However, there may be more local constraints than controls, or the matrix $A_j$ may be ill-conditioned. Hence, the factorization of $A_j$ should include pivoting and a rank decision to determine which rows can be used safely to eliminate control variables. The operations are as follows:

Factorize $A_j$, transform and partition $\Delta u_j^* \to (\Delta u_j^2, \Delta u_j^3)$, $\nu_j \to (\nu_j^u, \nu_j^x)$. Eliminate $\Delta u_j^2$ and $\nu_j^u$ formally. Factorize the resulting state constraint matrix $\bar{C}_j^x$, transform and partition $\Delta x_j^* \to (\Delta x_j^2, \Delta x_j^3)$. Determine $\Delta x_j^2$ immediately, eliminate

$\nu_j^x$ formally. As in step 2, eliminating $\Delta x_j^2$ splits matching conditions and yields partitioning $\lambda_{j-1}^* \to (\lambda_{j-1}^2, \lambda_{j-1}^3)$. The reduced local system can be written

$$\begin{bmatrix} H_j^{33} & {}^tJ_j^{33} & {}^tG_j^{33} \\ J_j^{33} & K_j^{33} & {}^tE_j^{33} \\ G_j^{33} & E_j^{33} & \end{bmatrix} \begin{bmatrix} \Delta x_j^3 \\ \Delta u_j^3 \\ \lambda_j^3 \end{bmatrix} + \begin{bmatrix} \tilde{f}_j^3 - \lambda_{j-1}^3 \\ \tilde{d}_j^3 \\ \tilde{h}_j^3 - \Delta x_{j+1}^3 \end{bmatrix} = 0, \tag{21}$$

$$S_{j+1}\Delta x_{j+1}^3 + [\alpha_{j+1} - \lambda_j^3] = 0. \tag{22}$$

**Steps 5 and 6.** *Project onto* $\ker\left(G_j^{33}\ E_j^{33}\ {-I}\right)$ *of projected matching conditions and compute Schur complement of projected control minimization conditions.* Eliminate $\Delta x_{j+1}^3, \lambda_j^3$ formally by substituting the third equation into the fourth one and the resulting expression into the first two equations. Eliminate $\Delta u_j^3$ formally. This completes recursion step $j$, leaving the projected state minimization condition which couples to stage $j-1$:

$$S_j \Delta x_j^3 + [\alpha_j - \lambda_{j-1}^3] = 0. \tag{23}$$

After the final recursion step 1 this “1-stage KKT system” is solved directly.

**Remarks.** In the case of second order decoupled boundary conditions the QP solver must deal with globally coupled constraints. Optimality of sub-arcs does not apply, and local minimization requires positive-definiteness of the Hessian on a larger space than $\ker(J_2^k)$. Projection steps 1 and 2 are not completely parallel in the right hand side transformation and solution phases. Fill-in is produced as Schur complement of the matrix of global constraints, and the recursion should include an additional SC step to exploit its sub-structure.

So far we assumed that the QP has a unique minimum. However, conflicting or rank-deficient constraints and semidefinite Hessians can be treated using more sophisticated factorizations like singular value and spectral decompositions.

# Interior point QP solution

Any phase inequality constraint appears in each stage of the discrete problem. Typical applications may include box constraints on all state and control variables. Efficient handling of that many inequalities remains a central difficulty.

In general we suggest a primal-dual barrier method as described below. However, an active set strategy with warm start may be competitive, especially when a good estimate of the correct active set is known and only a few exchange steps are expected. Note that large numbers of binding box constraints reduce the effort of recursive KKT solution, since such constraints are pre-eliminated in recursion steps 1 and 2, and that active set changes can be handled by a Schur complement method [5]. Improving a given solution by mesh refinement may therefore be very efficient with an active set strategy, and a good code should offer this alternative.

The concept of interior point or barrier methods for constrained optimization was introduced in [4] and theoretically developed in [3]. After the appearance of Karmarkar's paper [9] barrier methods for linear programs were developed by many authors, and extended to convex quadratic programs shortly thereafter. See, e.g., [13, 17, 7, 6]. A course classification of barrier methods both for LP and QP cases includes the primal, primal-dual, and predictor-corrector methods.

We adopt a primal-dual version of the QP barrier method described in [6], with the recursive KKT solver substituted. In the following the essentials of the approach to our problem formulation are sketched. Consider the general primal QP (9) with Lagrangian

$$L(y, w_2, w_3) = \frac{1}{2} y^T H y + J_1 y - (J_2 y + F_2)^T w_2 - (J_3 y + F_3)^T w_3 \tag{24}$$

(we drop superscript $k$ and write $y$ instead of $\Delta y$) and its dual problem

$$\max_{z, w_2, w_3} -\frac{1}{2} z^T H z - F_2^T w_2 - F_3^T w_3 \quad \text{s.t.} \quad \begin{cases} Hz + J_1^T - J_2^T w_2 - J_3^T w_3 = 0, \\ w_3 \geq 0. \end{cases} \tag{25}$$

Introducing slack variables $s$ we convert QP (9) to an equality-constrained barrier function form with potential parameter $\mu$:

$$\min_y \frac{1}{2} y^T H y + J_1 y - \mu \sum\nolimits_i \ln s_i \qquad \text{s.t.} \qquad \begin{cases} J_2 y + F_2 = 0, \\ -s + J_3 y + F_3 = 0. \end{cases} \tag{26}$$

With $S = \operatorname{diag}(\{s_i\})$, $W_3 = \operatorname{diag}(\{w_{3i}\})$, $e = (1, \ldots, 1)$, the first order necessary conditions can be written

$$\begin{aligned} Hy + J_1^T - J_2^T w_2 - J_3^T w_3 &= 0, & (27) \\ -S^{-1} \mu e + w_3 &= 0 \quad \text{or} \quad \mu e = S W_3 e, & (28) \\ J_2 y + F_2 &= 0, & (29) \\ -s + J_3 y + F_3 &= 0. & (30) \end{aligned}$$

A Newton step $(\delta y, \delta s, \delta w_2, \delta w_3)$ is obtained by solving the KKT system

$$\begin{pmatrix} H + J_3^T \Theta J_3 & J_2^T \\ J_2 & 0 \end{pmatrix} \begin{pmatrix} \delta y \\ -\delta w_2 \end{pmatrix} + \begin{pmatrix} (Hy + J_1^T - J_2^T w_2) - J_3^T \theta \\ J_2 y + F_2 \end{pmatrix} = 0 \tag{31}$$

and equations

$$\delta s = J_3 \delta y, \quad \delta w_3 = \theta - \Theta \delta s - w_3, \tag{32}$$

where $\Theta = S^{-1} W_3$, $\theta = S^{-1} \mu e$. Note that all iterates must satisfy equation (30) exactly, but need not satisfy (27,29). When $(y, s, w_2, w_3)$ satisfy the necessary conditions for any $\mu$, then $y$ is feasible for the primal QP, $(y, w_2, w_3)$ is feasible for the dual QP, and the duality gap is given by $s^T w_3 \geq 0$.

The primal equations are only slightly different. No information on the dual variable $w_3$ is maintained, and the definition of $\Theta$ must be changed to $\Theta = \mu S^{-2}$.

Mehrotra's predictor-corrector approach [12] defines $\hat{\theta} = 0$ first to compute the affine predictor step $(\delta\hat{y}, \delta\hat{s}, \delta\hat{w}_2, \delta\hat{w}_3)$; the corrector step as final search direction is then obtained for $\theta = S^{-1}(\mu e - \delta\hat{S}\delta\hat{W}_3 e)$.

We suggest the primal-dual approach because LP case experience of other authors shows that it is often more robust and takes fewer iterations than the primal approach, whereas the computational effort is only a little higher. On the other hand, a predictor-corrector method is considered superior only when matrix factorization clearly dominates the computations. This is not the case here, since any of the interior point variants respects our $m$-stage block-sparse QP structure:

**Theorem 2** *The modified Hessian $H + J_3^T \Theta J_3$ is block-diagonal.*

*Proof.* This follows since $\Theta$ is diagonal (for both definitions), and $H, J_3$ are block-diagonal according to theorem 1. ∎

Theoretical results indicate that a QP solution accuracy of $d$ digits can be achieved after $\mathcal{O}(\sqrt{N}d) = \mathcal{O}(\sqrt{m}d)$ iterations. Hence, the total effort would be $\mathcal{O}(m^{1.5}d)$ operations. It should be pointed out, however, that the $\sqrt{N}$ iteration count applies to arbitrary QP's of dimension $N$. But under certain regularity assumptions the switching structure of the solution of the nonlinear optimal control problem (i.e., the sequence of constraint arcs and free arcs) will be reproduced on any sufficiently fine grid. In that case we expect a better convergence behavior.

**Conjecture 1** *For a "regular" optimal control problem the QP subproblems are solved in $\mathcal{O}(d)$ interior point iterations and $\mathcal{O}(md)$ operations.*

# Conclusions

We have outlined a structured SQP method for nonlinear optimal control problems with path constraints. It was shown that a suitable choice of the discretization leads to QP subproblems resembling the very nature of the nonlinear optimal control problem in form of a specific $m$-stage block-sparse structure which is preserved when inequality constraints are treated by a primal-dual barrier method. A new recursive solver for the purely equality-constrained QP was presented which makes full use of the block-sparseness to produce a solution in $\mathcal{O}(m)$ operations.

# References

[1] J. T. Betts, W. P. Huffman: *Path Constrained Trajectory Optimization Using Sparse Sequential Quadratic Programming,* Boeing Computer Services, 1991.

[2] H. G. Bock, K.-J. Plitt: *A Multiple Shooting Algorithm for Direct Solution of Optimal Control Processes,* Proc. 9th IFAC World Congress, Budapest, 1984.

[3] A. V. Fiacco, G. P McCormick.: *Nonlinear Programming: Sequential Unconstrained Minimization Techniques,* John Wiley and Sons, New York, 1968.

[4] K. R. Frisch: *The Logarithmic Potential Method of Convex Programming,* Unpublished manuscript, University Institue of Economics, Oslo, 1955.

[5] P. E. Gill, W. Murray, M. A. Saunders, M. H. Wright: *A Schur-Complement Method for Sparse Quadratic Programming,* Report SOL 87-12, Department of Operations Research, Stanford University.

[6] P. E. Gill, W. Murray, D. B. Ponceleón, M. H. Wright: *Solving Reduced KKT Systems in Barrier Methods for Linear and Quadratic Programming,* Report SOL 91-7, Department of Operations Research, Stanford University.

[7] C. Gonzaga: *An Interior Trust Region Method for Linearly Constrained Optimization,* MPS Newsletter 19 (1991) 55–65.

[8] C. R. Hargraves, R. W. Paris: *Direct Trajectory Optimization Using Nonlinear Programming and Collocation,* AIAA J. Guidance 10 (1987) 338–342.

[9] N. Karmarkar: *A New Polynomial Time Algorithm for Linear Programming,* Combinatorica 4 (1984) 373–395.

[10] J. Konzelmann, H. G. Bock, R. W. Longman: *Time Optimal Trajectories of Polar Robot Manipulators by Direct Methods,* Modeling and Simulation, 20/5 (1989) 1933–1939, Instrument Society of America.

[11] J. Konzelmann, H. G. Bock, R. W. Longman: *Time Optimal Trajectories of Elbow Robots by Direct Methods,* Proc. AIAA Guidance, Navigation and Control Conference, Boston (1989) AIAA Paper 89-3530-CP.

[12] S. Mehrotra: *On the Implementation of a (Primal-Dual) Interior Point Method,* Technical Report 90-03, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 1990.

[13] R. D. C. Monteiro, I. Adler: *Interior Path Following Primal-Dual Algorithms. Part II: Convex Quadratic Programming,* Mathematical Programming 44 (1989) 43–66.

[14] K.-J. Plitt: *Ein superlinear konvergentes Mehrzielverfahren zur direkten Berechnung beschränkter optimaler Steuerungen*, Diploma Thesis (in German), Department of Applied Mathematics, University of Bonn, 1981.

[15] M. C. Steinbach, H. G. Bock, R. W. Longman: *Time Optimal Control of SCARA Robots,* Proc. AIAA Guidance, Navigation and Control Conference, Portland (1990) AIAA Paper 90-3394-CP.

[16] O. von Stryk: *Numerical Solution of Optimal Control Problems by Direct Collocation,* Report No. 322, Department of Mathematics, Munich University of Technology, 1991.

[17] Y. Ye, E. Tse: *An Extension of Karmarkar's Projective Algorithm for Convex Quadratic Programming,* Mathematical Programming 44 (1989) 157–179.