A Constraint Homotopy Active Set Solver for Linear-Quadratic Optimal Control

Johannes Buerger and Mark Cannon

***Abstract*—An efficient optimization method is proposed for linear-quadratic optimal control problems with state and control constraints. We describe an active set solver that uses Riccati recursions to solve a sequence of equality-constrained subproblems. The main contribution is a homotopy method based on relaxing inequality constraints. This overcomes known shortcomings of Riccati active set solvers relating to their initialisation and their application to problems with time-varying model data. It can be used exclusively or in combination with established Riccati active set solvers. The efficiency is demonstrated in numerical examples against state-of-the-art quadratic programming solvers.**

***Index Terms*—quadratic programming, linear-quadratic optimal control, active set solver, parametric programming**

I. Introduction

Driven by the great practical significance of linear-quadratic (LQ) optimal control problems, a range of efficient optimization methods have been proposed for LQ problems with inequality constraints on system states and control inputs. These solvers have made it possible to adopt constrained LQ optimal control and model predictive control (MPC) strategies in time-critical and safety-critical applications [1]. Generally, each optimization method has distinct, example-specific advantages.

Offline solution methods, such as explicit MPC [2], are attractive for small problems and horizon lengths. In this approach the feedback solution is stored as a piecewise affine function of the model state, which is treated as a parameter. However, whereas MPC solves an optimization problem for a given model state at each time-step, explicit MPC determines the solution for all states, resulting in high storage requirements, and moreover it relies on efficiently identifying online which affine feedback law applies to the current state.

For the majority of applications, online optimization methods are preferred to offline methods due to their lower storage and computational requirements. A common approach is to condense the problem formulation using the system model and employ a dense quadratic programming (QP) solver. In the context of MPC, a dual-feasible active set approach [3], [4] has proved popular and successful, however this does not exploit the LQ optimal control structure nor the parametric structure of the optimization, and its computation scales as the cube of the MPC horizon length due to the required initial matrix factorization. The primal-feasible active-set approach of [5] uses parametric properties (via homotopy) and can be warm-started. However it inherits the disadvantages of dense QP formulations in this context since it does not exploit the structure of the LQ optimal control problem.

As an alternative to dense formulations, QP solvers that explicitly exploit the optimal control structure have demonstrated significant computational benefits. Structured QP solvers typically use Riccati recursions [6], [7] to speed up the matrix factorizations at each iteration of an interior point or active set method. Generally, interior point methods can be made very efficient [7], [8], however, they are difficult to warm-start and can be numerically less reliable than active set methods. Active set methods, on the other hand, have proved successful due to their efficiency and numerical reliability [4], which are crucial properties in time-critical and safety-critical applications. Additionally, active set solvers are straightforward to warm-start, enabling further efficiency improvements in practice.

This note considers active set solvers that exploit both the parametric dependence and the LQ optimal control structure in the problem formulation. The main idea of exploiting parametric dependence (based on homotopy) was introduced in the context of nominal and robust MPC in [5], [9]–[11]. Starting from a given solution and its associated active set for a given model state, these methods trace active set changes during a line search from the given state towards the actual model state. The primal and dual solution trajectories are parameterized in terms of a homotopy parameter and thus the optimal solution at the current plant state is recovered. In [9]–[11] the equality constrained subproblems at each iteration are computed using Riccati recursions resulting in high computational efficiency. However, apart from special cases, it remains an open problem how to efficiently determine the initial solution for a given plant state. As a result, a computationally expensive Phase I procedure is generally required for initialization.

The proposed approach uses an initialization strategy similar to the dense QP solver [3], which relaxes constraints until the unconstrained solution is feasible. We use a homotopy parameter to scale inequality constraints, tracing the optimal solution of a relaxed problem towards that of the actual problem by performing active set updates to maintain dual feasibility. Each subproblem is solved using Riccati recursions, allowing efficient implementation with computation per iteration that depends linearly on the MPC horizon length. We use benchmark examples to illustrate computational performance.

II. Problem Statement

Consider the finite horizon LQ optimal control Problem 1:

$$\min_{\mathbf{u},\mathbf{x}} \sum_{k=0}^{N-1} \left(\tfrac{1}{2}\left\|u_k\right\|_{R_k}^2 + \tfrac{1}{2}\left\|x_k\right\|_{Q_k}^2\right) + \tfrac{1}{2}\left\|x_N\right\|_{Q_N}^2 \tag{1}$$

J. Buerger is with the BMW Group, Munich, Germany (e-mail: johannes.buerger@bmw.de).

M. Cannon is with the Department of Engineering Science, University of Oxford, U.K. (e-mail: mark.cannon@eng.ox.ac.uk).

$$\text{s.t. } \left.\begin{aligned} & x_{k+1} = A_k x_k + B_k u_k + d_k \\ & G_k u_k + F_k x_k \le \mathbf{1} \end{aligned}\right\} \text{ for } k = 0, \dots, N-1$$
$$x_0 = x^p.$$

Here N is the horizon length, Q_k, R_k are symmetric positive definite matrices (denoted $Q_k \succ 0$, $R_k \succ 0$), F_k and G_k define the inequality constraints, $\|v\|_S^2 = v^\top S v$ and $\mathbf{1} = [1 \cdots 1]^\top$. Additionally, x^p represents the current system state, $x_k \in \mathbb{R}^{n_x}$ and $u_k \in \mathbb{R}^{n_u}$ are the k steps ahead predicted state and control input, and $d_k \in \mathbb{R}^{n_x}$ is assumed known (e.g. a disturbance or a term resulting from a sequential QP formulation in the Nonlinear MPC context). Predicted control and state trajectories are denoted $\mathbf{u} = \{u_0, \dots, u_{N-1}\}$, $\mathbf{x} = \{x_0, \dots, x_N\}$.

Remark 1. Terminal state constraints are not explicitly included in Problem 1. However, constraints such as $F_N x_N \le \mathbf{1}$ can be enforced as equivalent constraints at stage $N-1$ as $G_{N-1}u_{N-1} + F_{N-1}x_{N-1} \le \mathbf{1}$, for appropriate matrices G_{N-1}, F_{N-1}. Note also that the parameters $Q_k, R_k, A_k, B_k, d_k, G_k, F_k$ for $k = 0, \dots, N-1$ and Q_N are assumed to be known but may vary between instants at which Problem 1 is solved. Problem 1 may be extended (with minor modifications to the development in Section III) to include general quadratic cost terms in $[x_k^\top \; u_k^\top]$ (leading to (u_k, x_k)-cross terms in the cost) and reference-tracking formulations based on the error terms $x_k - x_k^r$ and $u_k - u_k^r$, where x_k^r and u_k^r are the state and control input of a given reference trajectory.

Problem 1 can be solved using an active set method based on homotopy with x_0 as a parameter, e.g. [9], [11]. The approach requires knowledge of the optimal active set for a particular x_0 for initialisation. However, determining this can be a non-trivial task, and, except for special cases (such as problem formulations in which the optimal active set is empty when $x_0 = 0$), a costly Phase 1 procedure is needed. Therefore, it remains an open question how to design a general homotopy-based active set method employing Riccati recursions that overcomes this limitation and enables efficient application in the broad context of linear-quadratic optimal control. The target application areas include linear-quadratic MPC [1], [2], [8] and QP-based path planning [12], amongst many others.

This note performs a homotopy of the optimal solution of a related problem with an additional scalar parameter α. The corresponding problem formulation (generalizing Problem 1) is given (for some $\alpha \ge 1$) by Problem 2:

$$\min_{\mathbf{u},\mathbf{x}} \sum_{k=0}^{N-1} \left(\tfrac{1}{2}\|u_k\|_{R_k}^2 + \tfrac{1}{2}\|x_k\|_{Q_k}^2\right) + \tfrac{1}{2}\|x_N\|_{Q_N}^2 \tag{2}$$
$$\text{s.t. } \left.\begin{aligned} & x_{k+1} = A_k x_k + B_k u_k + d_k \\ & G_k u_k + F_k x_k \le \alpha\mathbf{1} \end{aligned}\right\} \text{ for } k = 0, \dots, N-1$$
$$x_0 = x^p.$$

Clearly, for $\alpha = 1$ this problem is identical to Problem 1. We exploit Problem 2 by continuously tracing the optimal solution for decreasing values of α until $\alpha = 1$, starting with $\alpha \ge 1$ such that no constraint is active. The approach is analogous to the classical dual-feasible method of Goldfarb-Idnani [3].

Remark 2. For general constraints $G_k x_k + F_k x_k \le b_k$, where b_k is any vector, we can express the scaled constraints as $G_k x_k + F_k x_k - b_k + \mathbf{1} \le \alpha\mathbf{1}$. The theory of Section III extends to this case with minor modifications.

III. Optimality Conditions and Riccati Recursion

Problem 2 can be solved using an equivalent stagewise problem formulation in which x_k, α are parameters:

$$\begin{aligned} L_k^*(x_k, \alpha) := \min_{u_k,\, x_{k+1}} \; & \tfrac{1}{2}(\|u_k\|_{R_k}^2 + \|x_k\|_{Q_k}^2) + L_{k+1}^*(x_{k+1}, \alpha) \\ \text{s.t. } & A_k x_k + B_k u_k + d_k = x_{k+1} \\ & G_k u_k + F_k x_k \le \alpha\mathbf{1} \end{aligned} \tag{3}$$

for $k = 0, \dots, N-1$, with initial condition $x_0 = x^p$ and terminal cost $L_N^*(x, \alpha) = \frac{1}{2}\|x\|_{Q_N}^2$. Given the scaling of constraints by α, the linear-quadratic structure of Problem 3 implies that the optimal value $L_k^*(x_k, \alpha)$ has the form

$$L_k^*(x_k, \alpha) = \tfrac{1}{2}\|x_k\|_{P_k}^2 + q_k^\top x_k + r_k, \tag{4a}$$
$$q_k = q_k^c + q_k^\alpha \alpha \tag{4b}$$

for $k = 0, \dots, N-1$, for some parameters P_k, r_k, q_k^c, q_k^α that depend on the active constraints at stage k.

Lemma 1. *$L_k^*(x, \alpha)$ is continuous, strictly convex, and piecewise quadratic in $x \in \mathcal{F}_k$, and $P_k \succ 0$ for all k.*

Proof. This can be shown by induction for $k = N-1, \dots, 0$ by a straightforward extension of [2] (Theorem 4 and Corollary 1) to Problem 3. In particular, if $L_{k+1}^*(x_{k+1}, \alpha)$ is continuous, convex, and piecewise quadratic in x_{k+1}, then since $Q_k \succ 0$, $R_k \succ 0$, the arguments of [2] imply that $L_k^*(x_k, \alpha)$ is continuous, strictly convex, and piecewise quadratic in x_k, and hence $P_k \succ 0$. Finally we note that $P_N = Q_N \succ 0$. $\square$

The optimality conditions for Problem 2 can be derived by defining a stagewise Lagrangian function associated with Problem 3 for $k = 0, \dots, N-1$ as

$$\begin{aligned} L_k(u_k, \lambda_k, \mu_k, x_{k+1}; x_k, \alpha) := \; & \tfrac{1}{2}\|u_k\|_{R_k}^2 + \tfrac{1}{2}\|x_k\|_{Q_k}^2 \\ & + \lambda_k^\top(A_k x_k + B_k u_k + d_k - x_{k+1}) \\ & + \mu_k^\top(G_k u_k + F_k x_k - \mathbf{1}\alpha) + L_{k+1}(x_{k+1}) \end{aligned}$$

where λ_k is the costate variable and μ_k the constraint multiplier at the kth stage (e.g. [11], Section 4).

The Karush–Kuhn–Tucker (KKT) necessary conditions for optimality (e.g. [13]) are given by, for $k = 0, \dots, N-1$,

$$\nabla_{u_k} L_k = R_k u_k + B_k^\top \lambda_k + G_k^\top \mu_k = \mathbf{0} \tag{5a}$$
$$\nabla_{\lambda_k} L_k = x_{k+1} - A_k x_k - B_k u_k - d_k = \mathbf{0} \tag{5b}$$
$$\begin{aligned} \nabla_{x_{k+1}} L_k &= -\lambda_k + \nabla_{x_{k+1}} L_{k+1} \\ &= -\lambda_k + Q_{k+1}x_{k+1} + A_{k+1}^\top \lambda_{k+1} + F_{k+1}^\top \mu_{k+1} = \mathbf{0} \end{aligned} \tag{5c}$$
$$\mu_k \ge \mathbf{0},\; \mu_k^\top(G_k u_k + F_k x_k - \alpha\mathbf{1}) = 0, \quad G_k u_k + F_k x_k - \alpha\mathbf{1} \le \mathbf{0} \tag{5d}$$

where $\mathbf{0} = [0 \cdots 0]^\top$.

Let $\mathcal{A}_k$ denote the active set at stage k, consisting of the indices of all active constraints at stage k of Problem 2:

$$\mathcal{A}_k := \{i : G_{k,i}u_k + F_{k,i}x_k = \alpha\},$$

where $G_{k,i}$, $F_{k,i}$ are the ith rows of G_k, F_k. Then

$$G_{k,a}u_k + F_{k,a}x_k = \alpha\mathbf{1} \tag{6}$$

where $G_{k,a}$, $F_{k,a}$ are the matrices formed by stacking rows $G_{k,i}$, $F_{k,i}$, $i \in \mathcal{A}_k$. Also define $\mu_{k,a}$ as the vector of elements of μ_k corresponding to active constraints and note that all other elements of μ_k are necessarily equal to zero due to (5d).

We consider separately the following two cases

case (a): $\operatorname{rank}([G_{k,a} \quad F_{k,a}]) = n_a$.
case (b): $\operatorname{rank}(G_{k,a}) = n_a$;

where n_a is the number of rows in $G_{k,a}$. Lemma 2 in Section V shows that case (a) always holds for the proposed active set algorithm, and hence case (b) holds if $F_k = 0$ or (trivially) if G_k has linearly independent rows.

From (5c) and (4) we have

$$\lambda_k = \nabla_{x_{k+1}} L_{k+1} = P_{k+1}x_{k+1} + q^c_{k+1} + q^\alpha_{k+1}\alpha. \tag{7}$$

Therefore conditions (5a), (5b), (5c) with k replaced by $k-1$, and (6) reduce to the linear system

$$K_k \begin{bmatrix} u_k \\ x_k \\ \mu_{k,a} \end{bmatrix} = \begin{bmatrix} 0 \\ \lambda_{k-1} \\ 0 \end{bmatrix} - \begin{bmatrix} B_k^\top q^\alpha_{k+1} \\ A_k^\top q^\alpha_{k+1} \\ -\mathbf{1} \end{bmatrix} \alpha - \begin{bmatrix} B_k^\top \\ A_k^\top \\ \mathbf{0} \end{bmatrix} (q^c_{k+1} + P_{k+1}d_k) \tag{8}$$

where K_k denotes the KKT matrix at stage k defined by

$$K_k = \begin{bmatrix} R_k + B_k^\top P_{k+1}B_k & B_k^\top P_{k+1}A_k & G_{k,a}^\top \\ A_k^\top P_{k+1}B_k & Q_k + A_k^\top P_{k+1}A_k & F_{k,a}^\top \\ G_{k,a} & F_{k,a} & \mathbf{0} \end{bmatrix}.$$

Theorem 1. *The optimal primal and dual variables of Problem 3 for a given active set $\mathcal{A}_k$ are given by (7) and*

$$u_k = H^u_k x_k + h^u_k + m^u_k\alpha \tag{9a}$$
$$\mu_{k,a} = H^\mu_k x_k + h^\mu_k + m^\mu_k\alpha \tag{9b}$$

and the optimal costate variable at stage $k-1$ is given by $\lambda_{k-1} = P_k x_k + q^c_k + q^\alpha_k\alpha$, where H^u_k, H^μ_k. h^u_k, h^μ_k, m^u_k, m^μ_k and P_k, q^c_k, q^α_k are uniquely determined by the solution of (8).

Proof. The KKT matrix in (8) has the 2×2-block structure

$$K_k = \begin{bmatrix} \Gamma_k & V_k^\top \\ V_k & 0 \end{bmatrix}, \tag{10}$$

$$\Gamma_k = \begin{bmatrix} R_k & 0 \\ 0 & Q_k \end{bmatrix} + \begin{bmatrix} B_k^\top \\ A_k^\top \end{bmatrix} P_{k+1} \begin{bmatrix} B_k^\top \\ A_k^\top \end{bmatrix}^\top, \ V_k^\top = \begin{bmatrix} G_{k,a}^\top \\ F_{k,a}^\top \end{bmatrix}.$$

Here $\Gamma_k \succ 0$ since $Q_k \succ 0$, $R_k \succ 0$ and $P_{k+1} \succ 0$ (by Lemma 1), and V_k has full (row) rank in either case (a) or case (b). Therefore K_k is non-singular (see e.g. [13], Thm 16.2). It follows that the solution of (8) can be expressed

$$u_k = S^u_k\lambda_{k-1} + s^u_k + t^u_k\alpha \tag{11a}$$
$$x_k = S^x_k\lambda_{k-1} + s^x_k + t^x_k\alpha \tag{11b}$$
$$\mu_{k,a} = S^\mu_k\lambda_{k-1} + s^\mu_k + t^\mu_k\alpha \tag{11c}$$

for a unique set of matrices S^u_k, S^x_k, S^μ_k and vectors s^u_k, s^x_k, s^μ_k, t^u_k, t^x_k, t^μ_k. But $P_k \succ 0$ (Lemma 1) implies $S^x_k \succ 0$, and hence

$$P_k = (S^x_k)^{-1}, \quad q^c_k = -(S^x_k)^{-1}s^x_k, \quad q^\alpha_k = -(S^x_k)^{-1}t^x_k \tag{12}$$

in (4). The proof is completed by using $\lambda_{k-1} = P_k x_k + q^c_k + q^\alpha_k\alpha$ to write u_k and $\mu_{k,a}$ in (11a,c) in the form of (9a,b). ☐

Remark 3. For case (b), the solution of Problem 3 at stage k can be obtained by solving a smaller KKT system. In particular, from (7), (5a), (5b), (6) we obtain the linear system

$$\begin{bmatrix} R_k + B_k^\top P_{k+1}B_k & G_{k,a}^\top \\ G_{k,a} & \mathbf{0} \end{bmatrix} \begin{bmatrix} u_k \\ \mu_{k,a} \end{bmatrix} = \\ -\begin{bmatrix} B_k^\top P_{k+1}A_k \\ F_{k,a} \end{bmatrix} x_k - \begin{bmatrix} B_k^\top q^\alpha_{k+1} \\ -\mathbf{1} \end{bmatrix} \alpha - \begin{bmatrix} B_k^\top \\ \mathbf{0} \end{bmatrix} (q^c_{k+1} + P_{k+1}d_k),$$

and the assumptions that $R_k \succ 0$ and $\operatorname{rank}(G_{k,a}) = n_a$ ensure that the reduced KKT matrix on the LHS of this equation is non-singular (e.g. [13], Thm 16.2). By solving this reduced KKT system we obtain the unique solutions for $u_k, \mu_{k,a}$ in the form (9). Hence (5c) with k replaced by $k-1$ implies

$$\begin{aligned} \lambda_{k-1} = {} & [Q_k + A_k^\top P_k(A_k + B_k H^u_k) + F_{k,a}^\top H^\mu_k]x_k \\ & + A_k^\top [P_k(B_k h^u_k + d_k) + F_{k,a}^\top h^\mu_k + q^c_k] \\ & + A_k^\top [P_k B_k m^u_k + F_{k,a}^\top m^\mu_k + q^\alpha_k]\alpha, \end{aligned}$$

which defines P_k, q^c_k, q^α_k satisfying $\lambda_{k-1} = P_k x_k + q^c_k + q^\alpha_k\alpha$ uniquely for the given active set $\mathcal{A}_k$.

Remark 4. For any given active set $\mathcal{A}_k$, the quadratic term in the optimal cost (4) is independent of α (due to the definition of Problem 2 with constraint scaling). This implies that only the offset terms in the state feedback law (9) depend on α.

Theorem 1 provides the solution of Problem 2 for a given sequence of active sets $\{\mathcal{A}_0, \ldots, \mathcal{A}_{N-1}\}$ through a backwards recursion requiring the solution of the KKT system (8) at each stage. The optimal primal and dual variables u_k, μ_k are affine functions of x_k and α. These relationships allow u_k, μ_k to be expressed in terms of the initial state x_0 and α using a forward simulation of the state trajectory, as discussed in Section IV.

IV. Forward Simulation and Geometric Interpretation

If the optimal control law $u_k = H^u_k x_k + h^u_k + m^u_k\alpha$ is known, then the system dynamics $x_{k+1} = A_k x_k + B_k u_k + d_k$ allow x_k to be determined as a function of x_0 and α in the form

$$x_k = \Phi_k x_0 + \phi^c_k + \phi^\alpha_k\alpha, \tag{13}$$

where Φ_k, ϕ^c_k, ϕ^α_k are defined by the recursive relationships

$$\Phi_{k+1} = (A_k + B_k H^u_k)\Phi_k \tag{14a}$$
$$\phi^c_{k+1} = (A_k + B_k H^u_k)\phi^c_k + B_k h^u_k + d_k \tag{14b}$$
$$\phi^\alpha_{k+1} = (A_k + B_k H^u_k)\phi^\alpha_k + B_k m^u_k \tag{14c}$$

with initial conditions $\Phi_0 = I$, $\phi^c_0 = 0$, $\phi^\alpha_0 = 0$. Using these expressions we can express the inactive primal inequality constraints, namely $G_{k,i}u_k + F_{k,i}x_k \le \alpha\mathbf{1}$ for $i \notin \mathcal{A}_k$, as

$$\begin{aligned} G_{k,i}u_k + F_{k,i}x_k &= G_{k,i}[H^u_k x_k + h^u_k + m^u_k\alpha] + F_{k,i}x_k \\ &= (G_{k,i}H^u_k + F_{k,i})(\Phi_k x_0 + \phi^c_k + \phi^\alpha_k\alpha) \\ &\quad + G_{k,i}(h^u_k + m^u_k\alpha) \le \alpha, \end{aligned} \tag{15}$$

for all $i \notin \mathcal{A}_k$, and the dual constraints, namely the non-negativity constraints on the multipliers $\mu_{k,a}$ of active inequality constraints, as

$$\begin{aligned}\mu_{k,a} &= H_k^\mu x_k + h_k^\mu + m_k^\mu \alpha \\ &= H_k^\mu(\Phi_k x_0 + \phi_k^c + \phi_k^\alpha \alpha) + h_k^\mu + m_k^\mu \alpha \geq \mathbf{0}. \qquad (16)\end{aligned}$$

Let $\Omega(\mathcal{A}, \alpha)$ denote the set of initial conditions x_0 satisfying the collection of primal constraints (15) and dual constraints (16) for $k = 0, \ldots, N-1$, for a given sequence of active sets

$$\mathcal{A} := \{\mathcal{A}_0, \ldots, \mathcal{A}_{N-1}\}.$$

Clearly $\Omega(\mathcal{A}, \alpha)$ is a convex polyhedral subset of the space of x_0 corresponding to a critical region [2] of Problem 2, when this is interpreted as a multiparametric programming problem with parameter x_0. Therefore the approach of this note can be interpreted as a generalization of known multiparametric QP approaches in the sense that α defines a homotopy of the original state space partition relative to the fixed ($\alpha = 1$) multiparametric programming regions corresponding to Problem 1.

V. Active Set Solver

We propose to solve Problem 1 by solving Problem 2 for a sequence of constraint scalings, $\{\alpha^0, \alpha^1, \ldots\}$, as described in Algorithm 1. Initially α^0 is determined so that the unconstrained solution is feasible for the current system state (i.e. $x^p \in \Omega(\emptyset, \alpha^0)$) by using the unconstrained Riccati recursion to compute the solution parameters in (9), (13). At the ith iteration the algorithm determines a new active set $\mathcal{A}^{i+1}$ by considering the smallest α such that $x^p \in \Omega(\mathcal{A}^i, \alpha)$. The procedure returns the solution to Problem 1 after a finite number of active set changes.

Algorithm 1: Riccati active set solver

Input : System state x^p, and parameters Q_N and $Q_k, R_k, A_k, B_k, d_k, G_k, F_k$, $k = 0, \ldots, N-1$.
Output: Optimal active set $\mathcal{A}^*$ and control input u^*.

1 Set $x_0 \leftarrow x^p$, $i \leftarrow 0$, $\mathcal{A}^i \leftarrow \emptyset$.
2 **while** $i = 0$ *or* $\alpha^i > 1$ **do**
3 for $k = N-1, \ldots, 0$ compute $H_k^u, h_k^u, m_k^u, H_k^\mu, h_k^\mu, m_k^\mu$ in (9),
4 for $k = 0, \ldots, N-1$ compute $\Phi_k, \phi_k^c, \phi_k^\alpha$ in (13) and the constraints in (15)-(16) defining $\Omega(\mathcal{A}^i, \alpha) \ni x_0$,
5 set $\alpha^i \leftarrow \min_{\alpha \geq 1}\{\alpha : x_0 \in \Omega(\mathcal{A}^i, \alpha)\}$, $\mathcal{A}^{i+1} \leftarrow \{\text{active set for } \alpha = \alpha^i\}$, $i \leftarrow i+1$.
6 Compute H_0^u, h_0^u, m_0^u in (9).
7 **return** $\mathcal{A}^* \leftarrow \mathcal{A}^i$ *and* $u^* \leftarrow H_0^u x^p + h_0^u + m_0^u$.

Remark 5. An efficient procedure for performing the line search $\alpha^i = \min_{\alpha \geq 1}\{\alpha : x_0 \in \Omega(\mathcal{A}^i, \alpha)\}$ in line 5 of Algorithm 1 is to find the values of α such that each individual inequality in (15), (16) holds with equality, and then set α^i equal to the largest of these values that is no less than 1. The required computation is proportional to the total number of constraints (e.g. $n_G N$ if G_k has n_G rows for all k). The active set is updated in line 5 by either adding or removing the constraint index corresponding to the smallest admissible value of α, depending on whether the constraint defining the critical value α^i belongs to (15) or (16), respectively.

Assumption 1. *Only a single constraint can become active or inactive at each iteration of Algorithm 1.*

Lemma 2. *At each iteration of Algorithm 1 the active set $\mathcal{A}^i$ satisfies* $\operatorname{rank}([G_{k,a}\ F_{k,a}]) = n_a$ *for* $k = 0, \ldots, N-1$.

Proof. This follows from the observation that the jth constraint in the inequality $G_k u_k + F_k x_k \leq \alpha \mathbf{1}$ cannot become active if $[G_{k,j}\ F_{k,j}]$ is linearly dependent on $[G_{k,a}\ F_{k,a}]$ since (6) necessarily holds. Hence the possibility of linear dependence in the active constraints is excluded by Assumption 1. □

Theorem 2. *Algorithm 1 converges to the optimal solution of Problem 1 in a finite number of iterations.*

Proof. With $\mathcal{A}^0 = \emptyset$, x^p necessarily belongs to $\Omega(\mathcal{A}^0, \alpha^0)$ for a sufficiently large but finite value of α^0. At each iteration $i \geq 0$, the line search over α in line 5 determines α^i and $\mathcal{A}^{i+1}$ such that $x^p \in \Omega(\mathcal{A}^i, \alpha^i) \cap \Omega(\mathcal{A}^{i+1}, \alpha^i)$, and the convex linear-quadratic structure of Problem 2 implies that $\mathcal{A}^{i+1} \neq \mathcal{A}^j$ for all $j \leq i$. The primal and dual variables in (9) are unique for each active set $\mathcal{A}^i$ (by Theorem 1), and their continuous dependence on α implies that i.e. $\alpha^{i+1} < \alpha^i$ for all i due to Assumption 1. Therefore the algorithm must converge within a finite number of iterations since the number of possible active sets and corresponding polyhedral regions is finite. □

Remark 6. The Riccati active set solver contains (for $\alpha = 1$) the Riccati Recursion employed in [9]–[11] as a special case. This fact can be exploited in a combined homotopy approach, where first a homotopy of constraint scalings (α) is used to obtain a solution for the current state x_a^p based on the approach of this note and then a homotopy of initial states (in the space of x_0) is performed whenever the model state is updated (to x_b^p), as outlined in [9]–[11]. This approach considers fixed critical regions $\Omega(\mathcal{A}, 1)$ and performs a homotopy from a model state x_a^p (with a known active set $\mathcal{A}^a$) towards the updated model state x_b^p (with updated active set $\mathcal{A}^b$). This is done by performing a line search in x_0-space via $x_0 = x_a^p + (x_b^p - x_a^p)\beta$, $\beta \in (0, 1]$). There is no direct link between α and β, since the homotopy parameters operate on separate quantities, but the intuition is as follows: β defines iterates of x_0 towards the current plant state and therefore defines a linear path through fixed critical regions and defines active set changes when x_0 intersects the boundary of each region, and α operates on the critical regions (by scaling them). Therefore the critical regions move through a fixed initial plant state $x_0 = x^p$. Thus active set changes occur whenever the initial plant state meets a boundary of the scaled critical region.

The approach of Remark 6 is favourable for time-invariant problem data as it uses the constraint scaling approach only in the initialization phase. For dynamic, time-varying problem structures (e.g. in sequential QP or QP-based path planning) it seems favourable to consider exclusively the approach of this note and hence avoid costly Phase I procedures.

Remark 7. A warm start strategy for Algorithm 1 may be implemented as follows. We replace line 1 by using the active set for the most recently computed solution as the initial guess of $\mathcal{A}^0$ at the current time. Then we check if $\alpha \geq 1$ exists such that the solution corresponding to this $\mathcal{A}^0$ is primal and dual feasible for the current system state x^p. If this is satisfied we proceed with the iteration in lines 2-7 of Algorithm 1. If no α exists such that the solution is primal and dual feasible, then active constraints have to be removed from $\mathcal{A}^0$ to recover primal and dual feasibility for some $\alpha \geq 1$, e.g. by removing the constraint corresponding to the most negative multiplier. In the worst case this may result in returning to unconstrained solution before making progress towards the actual solution.

VI. Numerical Properties

This section considers the computational effort required for each iteration of the proposed Riccati active set solver (Algorithm 1, lines 1-7). We assume that the linear subsystem (8) is solved using a LU decomposition method (see e.g. [13]) requiring $O(\frac{2}{3}(n_u + n_x + n_G)^3)$ floating point operations for case (a), or $O(\frac{2}{3}(n_u + n_G)^3)$ in case (b) (where n_G is the number of rows of G_k). The other significant contributions to computation in line 3 are the inversion of S^x_k by Cholesky factorization $O(\frac{1}{3}n_x^3)$ (case (a) only), and the matrix multiplications needed to compute the parameters in (9a,b), which require $O\big(2n_x^3 + n_x^2 n_G + n_x^2 n_u\big)$ operations. Combining these estimates for N stages gives an $O(N)$ dependence of computational complexity.

The computation required for the forward simulation in line 4 is $O(n_x^2 N)$ from (14a-c), and the computation of the line search and active set update (line 5) are comparatively insignificant. Overall therefore, the computation per iteration of Algorithm 1 grows as $O(N)$. This is in contrast to approaches such as [3], [5], [14] where the dense problem formulation has $O(N^3)$ complexity (as a result of the initial KKT factorizations in these algorithms). However, these algorithms often perform well for problems with short prediction horizons, since they efficiently update the initial KKT factorization using rank-one modifications after active set updates. We note that the matrices $H^u_k, h^u_k, m^u_k, H^\mu_k, h^\mu_k, m^\mu_k$ in (9) defining the solution presented in Section III only has to be updated for stages preceeding the specific stage $\hat{k}$ at which active set changes (and the solution for $k > \hat{k}$ can be re-used). This offers significant computational savings per iteration in practice (as demonstrated in Section VII) since an entire factorization update is very rarely required. Therefore it is an interesting future research direction to integrate the idea of low-rank modifications of Riccati factorizations into the proposed active set algorithm to obtain further computational improvements as proposed in [15] for input-constrained systems.

VII. Simulation Examples

A. Example 1

We consider a linear time-varying (LTV) optimal control problem which results from a lateral trajectory planning formulation where the corresponding input is the rate of change of the desired curvature relative to a given reference path (for details see e.g. [12]). The system has one real input u_k and three slack variable inputs $e_{k,2}, e_{k,3}, e_{k,4}$ per stage (i.e. they do not affect the system dynamics). The approach of [12] first computes a longitudinal trajectory resulting in longitudinal velocities v_k ($k = 0, \dots, N-1$). In a second step the lateral trajectory planning problem is formulated based on the values of v_k (as time-varying system parameters) and the system description is given by:

Solver	Riccati AS	HPIPM	Goldfarb-Idnani AS	Quadprog IP(dense)
CPU Time (ms) Iterations	1.3/2.7 9.1/17	1.7/2.9 7.2/10	6.8/8.7 8.4/15	41.1/49.0 6.5/9

TABLE I: Example 1 mean/max solution time (row 1) and mean/max number of iterations (row 2)

$$A_k = \begin{bmatrix} 1 & 0.2v_k & 0.02v_k^2 & 0.0013v_k^2 \\ 0 & 1 & 0.2v_k & 0.02v_k^2 \\ 0 & 0 & 1 & 0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_k = \begin{bmatrix} 0.0007v_k^2 & 0 & 0 & 0 \\ 0.0013v_k & 0 & 0 & 0 \\ 0.02 & 0 & 0 & 0 \\ 0.2 & 0 & 0 & 0 \end{bmatrix}.$$

The cost is defined by $Q = \mathrm{diag}\{0.1, 0.1, 0.1, 0.1\}$, $R = \mathrm{diag}\{100, 1, 1, 1\}$ (where R contains additonal weights for the slack variable inputs). The constraints are given by:

$$\begin{aligned} -0.1 \leq u_k &\leq 0.1 \\ e_{k,i} &\geq 0 && \text{for } i = 2,3,4 \\ -0.2 - e_{k,i} \leq x_{k,i} &\leq 0.2 + e_{k,i} && \text{for } i = 2,3,4 \end{aligned}$$

and the horizon length is set to $N = 30$. Algorithm 1 was implemented in Matlab [16], with computationally intensive parts implemented via C-mex functions. In this case the Riccati solver is benchmarked for initial conditions $x_0 = [r \ 0 \ 0 \ 0]^\top$ with randomly selected $r \in [0, 10]$ (over 1000 trials). Table I shows the computational performance of Matlab's (dense) interior point (IP) solver [16], the Goldfarb-Idnani active set (AS) method [3], the HPIPM solver [7] and the Riccati AS algorithm of this note (using an AMD Ryzen 7 PRO 4750U, 1.7 GHz processor) for cold starts. We note that the Riccati AS and HPIPM solver show similar computational performance for cold starting on Example 1. We further consider warm starting of the Riccati AS solver by simulating a closed loop system over 15 time steps with an initial plant state given by $x_0 = [10 \ 0 \ 0 \ 0]^\top$. The Riccati AS solver then requires on average $0.5\,$ms (3 iterations) whereas HPIPM requires $2.2\,$ms (7.5 iterations), demonstrating the significant computational gains of warm starting in combination with the proposed method. We remark that in Section VII we have chosen comparisons with established QP solvers (such as the optimal control structure-exploiting interior point solver HPIPM and the Goldfarb-Idnani active set method, which is widely regarded as efficient for condensed MPC formulations). However, we do not consider active set solvers requiring specialized phase I procedures for obtaining an initial solution (such as the Riccati active set method [9]–[11]).

B. Example 2

We consider the effect of varying the horizon N of an MPC strategy with time-invariant model parameters

$$A = \begin{bmatrix} 0.94 & 0.13 & 0 & 0 \\ -0.54 & 0.4 & 0 & 0 \\ 0.12 & 0.007 & 1 & 0.2 \\ 1.17 & 0.09 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0.05 & 1 & 0 & 0 \\ 0.4 & 0 & 1 & 0 \\ -0.02 & 0 & 0 & 0 \\ -0.18 & 0 & 0 & 1 \end{bmatrix}.$$

and with cost weights $Q = \mathrm{diag}\{0.5, 1, 1.5, 2\}$ and $R = \mathrm{diag}\{0.25, 0.25, 0.25, 0.25\}$. The constraints are given by

$$-25 \leq u_{k,i} \leq 25 \quad \text{for } i = 1,2,3,4.$$

Solver Time (ms) and Iterations	$N = 20$	$N = 30$	$N = 40$	$N = 50$
Riccati AS	1.0/1.7	1.4/2.4	1.8/3.3	2.0/3.7
	8.1/17	9.9/18	9.2/19	8.9/17
HPIPM	1.5/2.3	2.1/3.1	2.8/4.1	3.4/5.1
	8.2/12	7.7/11	7.9/11	7.5/11
Goldfarb-Idnani AS	8.2/10.1	13.3/6.9	24.1/28.7	27.8/38.3
	9.1/15	8.3/17	8.1/18	9.2/19
Quadprog IP (dense)	6.9/7.1	8.3/8.5	9.3/10.0	11.4/12.7
	7.1/8	6.3/7	6.6/7	6.3/7

TABLE II: Example 2 mean/max solution time (row 1) and mean/max number of iterations (row 2)

Solver Time (ms) and Iterations	$N = 20$	$N = 30$	$N = 40$	$N = 50$
Riccati AS	0.3/0.6	0.4/0.5	0.5/0.6	0.6/0.8
	3.9/9	3.7/7	4.2/9	3.8/8
HPIPM	2.1/3.9	2.4/3.7	2.6/4.3	3.2/5.7
	13.2/20	10.9/18	9.8/16	9.6/17
Goldfarb-Idnani AS	2.6/3.7	2.9/3.8	3.3/4.2	3.6/4.5
	3.2/7	4.3/8	5.1/10	4.2/6
Quadprog IP (dense)	5.3/6.3	5.5/5.8	5.5/6.1	5.8/6.4
	10.2/12	9.5/11	9.1/12	8.1/11

TABLE III: Example 3 mean/max solution time (row 1) and mean/max number of iterations (row 2)

Benchmark tests were performed for initial conditions $x_0 = [0\ 0\ r\ 0]^\top$ with $r \in [0, 150]$ selected randomly (1000 trials). The results for varying horizon length N are given in Table II.

C. Example 3

We consider an MPC problem with varying N, with time-invariant model parameters

$$A = \begin{bmatrix} 0.94 & 0.13 & 0 & 0 \\ -0.54 & 0.4 & 0 & 0 \\ 0.12 & 0.007 & 1 & 0.2 \\ 1.17 & 0.09 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0.05 \\ 0.4 \\ -0.02 \\ -0.181 \end{bmatrix}.$$

The cost weights are given by $Q = \mathrm{diag}\{0.5, 1, 1.5, 2\}$, $R = 0.25$, and the constraints are

$$-25 \le u_k \le 25.$$

Benchmark tests were performed for initial conditions $x_0 = [0\ 0\ r\ 0]^\top$ with $r \in [0, 150]$ selected randomly (1000 trials). The results for varying horizon length N are given in Table III.

D. Example 4

We consider a time-invariant MPC problem with $N = 30$, $n_x = 12$, $n_u = 4$ defined by $A = \mathrm{diag}\{A_2, 0.5A_2, 0.25A_2\}$, $B = [B_2^\top, 0.5B_2^\top, 0.25B_2^\top]^\top$, where A_2 and B_2 correspond to the system in Example 2, $Q = 0.5I$, $R = 0.25I$ and constraints $-25 \le u_{k,i} \le 25 \quad$ for $i = 1, 2, 3, 4$. Initial conditions are $x_0 = [0\ 0\ r\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^\top$ with $r \in [0, 350]$ selected randomly (1000 trials). The results are given in Table IV.

Solver	Riccati AS	HPIPM	Goldfarb-Idnani AS	Quadprog IP(dense)
CPU Time (ms)	5.2/9.2	6.8/9.0	12.5/16.4	8.7/10.6
Iterations	9.6/25	9.4/14	9.2/23	8.7/14

TABLE IV: Example 4 mean/max solution time (row 1) and mean/max number of iterations (row 2)

In summary, each of these numerical examples demonstrates the significant potential of the proposed algorithm over generic state-of-the-art QP solvers. The improved scaling of computation time with problem size, in particular horizon length N, illustrate its effectiveness for trajectory planning and receding horizon control applications.

VIII. Conclusion

This note addresses the issue of how to efficiently initialize a Riccati active set solver. Finite-time convergence to the solution is established and per-iteration computation bounds are derived. Computational efficiency is demonstrated using numerical examples. The proposed framework has potential future extensions in the context of QP-based path-planning applications [12], nonlinear MPC using successive linearisation [17], and sequential QP strategies for nonlinear MPC [18].

References

[1] D. Kouzoupis, A. Zanelli, H. Peyrl, and H. Ferreau, "Towards proper assessment of QP algorithms for embedded model predictive control," in *2015 European Control Conference (ECC)*, pp. 2609–2616, 2015.

[2] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, pp. 3–20, 2002.

[3] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical Programming*, vol. 27, pp. 1–33, 1983.

[4] G. Cimini and A. Bemporad, "Exact complexity certification of active-set methods for quadratic programming," *IEEE Transactions on Automatic Control*, vol. 62, no. 12, pp. 6094–6109, 2017.

[5] H. Ferreau, H. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *Int. J. Robust Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.

[6] A. Bryson and Y.-C. Ho, *Applied optimal control*. Hemisphere, 1975.

[7] G. Frison and M. Diehl, "HPIPM: a high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020. 21st IFAC World Congress.

[8] C. Rao, S. Wright, and J. Rawlings, "Application of interior-point methods to model predictive control," *Journal of Optimization Theory and Applications*, vol. 99, pp. 723–757, 1998.

[9] M. Cannon, W. Liao, and B. Kouvaritakis, "Efficient MPC optimization using Pontryagin's Minimum Principle," *Int. J. Robust Nonlinear Control*, vol. 18, no. 8, pp. 831–844, 2008.

[10] J. Buerger, M. Cannon, and B. Kouvaritakis, "An active set solver for input-constrained robust receding horizon control," *Automatica*, vol. 1, pp. 155–161, 2014.

[11] J. Buerger, M. Cannon, and B. Kouvaritakis, "Active set solver for min-max robust control with state and input constraints," *Int. J. Robust Nonlinear Control*, vol. 26, no. 15, pp. 3209–3231, 2016.

[12] B. Gutjahr, L. Gröll, and M. Werling, "Lateral vehicle trajectory optimization using constrained linear time-varying MPC," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1586–1595, 2017.

[13] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, 2nd ed., 2006.

[14] D. Arnstrom, A. Bemporad, and D. Axehill, "A dual active-set solver for embedded quadratic programming using recursive LDL' updates," *IEEE Transactions on Automatic Control*, vol. 67, no. 8, pp. 4362 – 4369, 2022.

[15] I. Nielsen and D. Axehill, "Low-rank modifications of Riccati factorizations for model predictive control," *IEEE Transactions on Automatic Control*, vol. 63, no. 3, pp. 872–879, 2017.

[16] MATLAB, *R2022a*. Natick, Massachusetts: The MathWorks Inc., 2022.

[17] M. Cannon, J. Buerger, B. Kouvaritakis, and S. Rakovic, "Robust tubes in nonlinear model predictive control," *IEEE Trans. Automatic Control*, vol. 56, no. 8, pp. 1942–1947, 2011.

[18] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, "From linear to nonlinear MPC: bridging the gap via the real-time iteration," *International Journal of Control*, vol. 93, no. 1, pp. 62–80, 2020.