MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Energy-Efficient Collision-Free Trajectory Planning Using Alternating Quadratic Programming

Zhao, Y.; Wang, Y.; Bortoff, S.A.; Nikovski, D.

TR2014-046 June 2014

Abstract

This paper considers the planning of collision-free and energy-optimal trajectories for linear systems with decoupled dynamics for different degrees of freedom. A direct transcription of such a problem generally results in a non-convex problem due to the collision avoidance constraint. In this paper we propose a novel Alternating Quadratic Programming (AQP) algorithm to deal with the non-convex collision avoidance constraint, and generate a suboptimal solution for the original problem by alternatively solving a number of subproblems. It is proved that the AQP algorithm is guaranteed to converge, and the solution is locally optimal when the boundary of the collision-free region satisfies certain properties. The speed and energy-saving performance of the proposed method are demonstrated by numerical examples.

American Control Conference (ACC), 2014

Energy-Efficient Collision-Free Trajectory Planning Using Alternating Quadratic Programming

Yiming Zhao, Yebin Wang, Scott A. Bortoff and Daniel Nikovski

Abstract— **This paper considers the planning of collision-free and energy-optimal trajectories for linear systems with decoupled dynamics for different degrees of freedom. A direct transcription of such a problem generally results in a non-convex problem due to the collision avoidance constraint. In this paper we propose a novel Alternating Quadratic Programming (AQP) algorithm to deal with the non-convex collision avoidance constraint, and generate a suboptimal solution for the original problem by alternatively solving a number of subproblems. It is proved that the AQP algorithm is guaranteed to converge, and the solution is locally optimal when the boundary of the collision-free region satisfies certain properties. The speed and energy-saving performance of the proposed method are demonstrated by Numerical examples.**

Index Terms— **motion planning, trajectory generation, minimum energy, collision avoidance, Quadratic Programming.**

I. Introduction

This paper focuses on the planning of energy-optimal collision-free trajectories for a special type of linear systems where the dynamics for different degrees-of-freedom (DOF) are decoupled. Examples of such systems include 3D printer, gate crane, and satellite (translational dynamics only), etc. In order to address various constraints such as collision avoidance, speed and control constraints, we explore the numerical optimization approach, which is more tractable for dealing with constraints than theoretical optimal control.

The collision-free trajectory planning problem is typically solved by decomposing it into two levels. At the higher level, only the geometric aspects of the path including the collision avoidance constraint are considered, while the lower level deals with the system dynamics and other constraints, and either filters the path provided by the higher (geometric) level planner into a collision-free trajectory [6], or assigns a speed profile along the path to transform it into a trajectory [2], [11], [10], [12], [13]. Such a decomposation approach is usually computationally efficient. However, it may fail to find a feasible trajectory if the geometric path is not properly generated. Neither does it ensure the optimality of the planned trajectory.

The collision-free trajectory planning problem has also been solved in the full state space directly using Mixed-Integer Linear Programming (MILP)[8], and Mixed-Integer Quadratic Programming (MIQP)[7] for linear dynamical systems with linear and quadratic cost functions. While these methods can find the globally optimal solution, the problem scale and computation time grow substantially when the topology of the collision-free region is complex.

In this paper, we explore a different direction to generate collision-free energy-optimal trajectories for a type of linear systems. In particular, we take advantage of the decoupled nature of system dynamics and the cost function to generate an energy-efficient collision-free trajectory by iteratively solving a number of convex subproblems. Such a method is implemented as an algorithm called the Alternating Quadratic Programming algorithm, since the subproblems are formulated as Quadratic Programming (QP) problems and solved alternatively. By solving each subproblem, the energy efficiency of the system's motion in the DOF corresponding to this particular subproblem is improved, while ensuring the feasibility of the trajectory, i.e., satisfying all constraints including the system dynamics. It is proved that the proposed method is guaranteed to converge. Furthermore, when the boundary of the non-convex collision-free region satisfies certain conditions, the AQP algorithm always generates a locally optimal solution. This method is both computationally efficient and reliable, hence is suitable for real-time applications.

The organization of this paper is as follows: We first present the system dynamics and constraints, and show that the original non-convex cost function is convex in the subspace defined by the system dynamics. Then we discretize the energy-optimal collision-free trajectory generation problem minimizing the equivalent convex cost function, and introduce the AQP algorithm in Section III. The convergence of the AQP algorithm and the optimality of the solution are analyzed. Finally, numerical examples are given in Section IV to validate the effectiveness of the AQP algorithm.

II. Convexity Analysis and Problem Formulation

A. System Dynamics

For simplicity, we consider a linear dynamical system with two decoupled translational DOF as an example to introduce the problem formulation and present the proposed method, although the results can be extended to more general systems with decoupled motions. In particular, the dynamics of the system has the following form

$$\dot{x} = v_x, \quad (1)$$

$$\dot{y} = v_y, \quad (2)$$

$$\dot{v}_x = -d_x v_x + b_x u_x, \quad (3)$$

$$\dot{v}_y = -d_y v_y + b_y u_y, \quad (4)$$

Y. Zhao is with Halliburton Energy Service, 3000 N Sam Houston Pkwy E, Houston,TX,77032, yiming.zhao@halliburton.com; Y. Wang, S. A. Bortoff, and D. Nikovski are with Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA, 02139, USA. {yebin.wang, bortoff,nikovski}@merl.com

where $x, y \in \mathbb{R}$ are the position of the system, $v_x, v_y \in \mathbb{R}$ are the speed of the system along the x and y directions, respectively. u_x, u_y are the control inputs, $d_x, d_y > 0$ are friction coefficients, and $b_x, b_y > 0$ are constants. The system is required to start from position (x_0, y_0) at time $t = 0$ with initial speed (v_{x_0}, v_{y_0}), and move to (x_f, y_f) at time $t = t_f$ with final speed (v_{x_f}, v_{y_f}).

B. Cost Function and Constraints

We consider the minimization of a quadratic energy consumption cost function:

$$\begin{aligned} J(v_x, u_x, v_y, u_y) &= J_x(v_x, u_x) + J_y(v_y, u_y) \\ &= \int_0^{t_f} \left(R_x u_x^2 + K_x v_x u_x + R_y u_y^2 + K_y v_y u_y\right) \mathrm{d}t \end{aligned} \quad (5)$$

Such a cost function is commonly used in the literature as an estimation of the energy consumption of mechatronic systems, where the $R_x u_x^2$ and $R_y u_y^2$ terms inside the integral of (5) correspond to the copper loss, and the $K_x v_x u_x$ and $K_y v_y u_y$ terms are the mechanical power. The constants R_x, R_y, K_x, K_y are positive numbers. The summation of these terms approximates the instantaneous power consumption of the system.

The representation of obstacles is an important step for the formulation of collision-free trajectory generation problem. To facilitate problem formulation, regular shapes such as polytopes, cylinders, and spheres are often used to describe the obstacles [5], [9], [8], or alternatively, the feasible (obstacle-free) region or volume for the system's motion [3]. In this paper, we use the feasible region formulation, and assume that the time-invariant obstacle-free region is represented by a closed set $\mathcal{D} \subset \mathbb{R}^2$.

The motion of the system must satisfy:

1) dynamics constraints (1) to (4),
2) the boundary conditions $x(0) = x_0$, $x(t_f) = x_f$, $v_x(0) = v_{x_0}$, $v_x(t_f) = v_{x_f}$, $y(0) = y_0$, $y(t_f) = y_f$, $v_y(0) = v_{y_0}$, $v_y(t_f) = v_{y_f}$.
3) the speed constraints

$$v_{x_{\min}} \le v_x(t) \le v_{x_{\max}} \quad (6)$$
$$v_{y_{\min}} \le v_y(t) \le v_{y_{\max}}, \quad t \in [0, t_f] \quad (7)$$

4) the control constraints

$$u_{x_{\min}} \le u_x(t) \le u_{x_{\max}} \quad (8)$$
$$u_{y_{\min}} \le u_y(t) \le u_{y_{\max}}, \quad t \in [0, t_f] \quad (9)$$

5) and the collision-avoidance constraint

$$(x(t), y(t)) \in \mathcal{D}, \quad t \in [0, t_f], \quad (10)$$

where it is assumed that $(x_0, y_0) \in \mathcal{D}$ and $(x_f, y_f) \in \mathcal{D}$. The inequalities above are taken componentwise. One remarkable feature of the systems considered in this paper is that both the energy consumption and the dynamics are decoupled for different translational motions. As a result, when the obstacle avoidance constraint (10) is not active, the generation of trajectory in the two dimensional plane can be accomplished by planning the motions along the x and y directions separately. In this case, each separated motion planning problem can be solved efficiently using numerical optimization. In the presence of obstacles, however, the two subproblems are coupled via the collision avoidance constraint (10), which is non-convex. In the following, we will develop a numerical optimization method to efficiently handle such a case when (10) is active.

C. Convexification of the Cost Function

It is well-known that the convexity of the problem, which includes the convexity of the cost function and the convexity of the feasible region, is one of the main factors affecting the convergence and the efficiency of numerical optimization. It is obvious that the cost function (5) is non-convex in v_x, u_x, v_y and u_y. Although the optimal control problem for the minimization of (5) subject to constraints (1) to (10) can be solved directly using numerical optimization, the convergence is not ensured in general for such a non-convex cost function. However, as will be shown below, more insights can be obtained by analyzing the convexity of (5), such that the problem can be solved with better numerical efficiency and reliability.

It is noted that (5) can be written equivalently as a convex function of v_x, u_x, v_y, and u_y considering (3) and (4). In particular, multiplying both sides of (3) by v_x, we have:

$$v_x \dot{v}_x = -d_x v_x^2 + b_x u_x v_x,$$

Re-arranging the terms of the above expression and integrating both sides on $[0, t_f]$, we have

$$\begin{aligned} &\int_0^{t_f} K_x u_x v_x \, \mathrm{d}t \\ &= \int_0^{t_f} \frac{K_x}{b_x} \left(v_x \dot{v}_x + d_x v_x^2\right) \mathrm{d}t \\ &= \frac{K_x}{b_x} \left(\int_0^{t_f} v_x \dot{v}_x \, \mathrm{d}t + \int_0^{t_f} d_x v_x^2 \, \mathrm{d}t\right) \\ &= \frac{K_x}{b_x} \left(\int_0^{t_f} \frac{1}{2} \, \mathrm{d}v_x^2 + \int_0^{t_f} d_x v_x^2 \, \mathrm{d}t\right) \\ &= \frac{K_x}{b} \left(\frac{1}{2} v_x^2(t_f) - \frac{1}{2} v_x^2(0) + d_x \int_0^{t_f} v_x^2 \, \mathrm{d}t\right) \\ &= \frac{d_x K_x}{b_x} \int_0^{t_f} v_x^2 \, \mathrm{d}t + C_x \end{aligned}$$

where C_x is a constant determined by the boundary conditions. Similarly, the mechanical power consumed by the motion in the y direction can be written as

$$\int_0^{t_f} K_y u_y v_y \, \mathrm{d}t = \frac{d_y K_y}{b_y} \int_0^{t_f} v_y^2 \, \mathrm{d}t + C_y.$$

Let $Q_x = d_x K_x / b_x$, and $Q_y = d_y K_y / b_y$, which are positive numbers. Then (5) can be rewritten as

$$\int_0^{t_f} \left(R_x u_x^2 + Q_x v_x^2 + R_y u_y^2 + Q_y v_y^2\right) \mathrm{d}t + C_x + C_y$$

Since C_x and C_y are constants, it is equivalent to minimize the following cost function instead of (5):

$$\tilde{J}(v_x, u_x, v_y, u_y)$$

$$= \tilde{J}_x(v_x, u_x) + \tilde{J}_y(v_y, u_y)$$
$$= \int_0^{t_f} \left(R_x u_x^2 + Q_x v_x^2 + R_y u_y^2 + Q_y v_y^2\right) \mathrm{d}t$$

which is convex in v_x, u_x, v_y, u_y. The definitions of $\tilde{J}_x$ and $\tilde{J}_y$ are obvious from the above expression.

D. Problem Discretization

We discretize both the cost function and system dynamics on a mesh in the time domain using the trapezoidal integration rule. Let the time grid (possibly non-uniform) be $\{t_i\}_{i=0}^N \in [t_0, t_f]$, with $t_N = t_f$. Also, let $\Delta_i = t_i - t_{i-1}$ for any $i = 1, \ldots, N$. The state and control variables are discretized on the mesh $\{t_i\}_{i=0}^N$ as follows: $x_i = x(t_i)$, $v_{x_i} = v_x(t_i)$, $y_i = y(t_i)$, $v_{y_i} = v_y(t_i)$ for $i = 0, \ldots, N$, and $u_{x_i} = u_x(\frac{t_i + t_{i-1}}{2})$, $u_{y_i} = u_y(\frac{t_i + t_{i-1}}{2})$ for $i = 1, \ldots, N$. For notational convenience, let $X = [x_0, x_1, \ldots, x_N]^T$, $V_x = [v_{x_0}, v_{x_1}, \ldots, v_{x_N}]^T$, $U_x = [u_{x_1}, u_{x_2}, \ldots, u_{x_N}]^T$, $Y = [y_0, y_1, \ldots, y_N]^T$, $V_y = [v_{y_0}, v_{y_1}, \ldots, v_{y_N}]^T$, $U_y = [u_{y_1}, u_{y_2}, \ldots, u_{y_N}]^T$, $\mathcal{X} = [V_x^T, U_x^T]^T$, $\mathcal{Y} = [V_y^T, U_y^T]^T$.

The cost function $\tilde{J}$ is discretized similarly using the trapezoidal integration rule as

$$\begin{aligned}\tilde{J}(\mathcal{X}, \mathcal{Y}) &= \tilde{J}_x(\mathcal{X}) + \tilde{J}_y(\mathcal{Y}) \\ &= \sum_{i=1}^N \Delta_i \left(R_x U_{x_i}^2 + \frac{Q_x}{2} V_{x_{i-1}}^2 + \frac{Q_x}{2} V_{x_i}^2\right) \\ &+ \sum_{i=1}^N \Delta_i \left(R_y U_{y_i}^2 + \frac{Q_y}{2} V_{y_{i-1}}^2 + \frac{Q_x}{2} V_{y_i}^2\right)\end{aligned} \tag{11}$$

The system dynamics (1) to (4) are enforced between neighboring grid points by the following linear equations using the trapezoidal integration rule:

$$2(X_i - X_{i-1}) = \Delta_i (V_{x_i} + V_{x_{i-1}}) \tag{12}$$
$$2(Y_i - Y_{i-1}) = \Delta_i (V_{y_i} + V_{y_{i-1}}) \tag{13}$$
$$2(V_{x_i} - V_{x_{i-1}}) = \Delta_i \left(-d_x(V_{x_i} + V_{x_{i-1}}) + 2b_x U_{x_i}\right) \tag{14}$$
$$2(V_{y_i} - V_{y_{i-1}}) = \Delta_i \left(-d_y(V_{y_i} + V_{y_{i-1}}) + 2b_y U_{y_i}\right) \tag{15}$$

where $(\cdot)_i$ denotes the i^{th} component of a vector for $i = 1, 2, \ldots, N$. The discretized speed and control constraints are given by

$$v_{x_{\min}} \le V_x \le v_{x_{\max}} \tag{16}$$
$$v_{y_{\min}} \le V_y \le v_{y_{\max}} \tag{17}$$
$$u_{x_{\min}} \le U_x \le u_{x_{\max}} \tag{18}$$
$$u_{y_{\min}} \le U_y \le u_{y_{\max}} \tag{19}$$

The other constraints include the initial and final conditions

$$[X_1, X_{N+1}] = [x_0, x_f] \tag{20}$$
$$[Y_1, Y_{N+1}] = [y_0, y_f] \tag{21}$$
$$[V_{x_1}, V_{x_N}] = [v_{x_0}, v_{x_f}] \tag{22}$$
$$[V_{y_1}, V_{y_N}] = [v_{y_0}, v_{y_f}] \tag{23}$$

and the collision-avoidance constraint

$$(X_i, Y_i) \in \mathcal{D}, \quad i = 1, \ldots, N+1. \tag{24}$$

In order to reduce the number of decision variables, we eliminate parameters X_i and Y_i from the boundary conditions (20), (21) and the collision-avoidance constraint (24) using the (12) and (13), and obtain the following equivalent conditions:

$$\frac{1}{2} \sum_{i=1}^N \Delta_i \left(V_{x_i} + V_{x_{i-1}}\right) = x_f - x_0 \tag{25}$$
$$\frac{1}{2} \sum_{i=1}^N \Delta_i \left(V_{y_i} + V_{y_{i-1}}\right) = y_f - y_0 \tag{26}$$

$$\begin{aligned}&\left(\frac{1}{2} \sum_{i=1}^k \Delta_i \left(V_{x_i} + V_{x_{i-1}}\right), \frac{1}{2} \sum_{i=1}^k \Delta_i \left(V_{y_i} + V_{y_{i-1}}\right)\right) \\ &+(x_0, y_0) \in \mathcal{D}, \quad k = 1, \ldots, N-1.\end{aligned} \tag{27}$$

Note that the collision avoidance constraint is not enforced at the first and the last nodes since the initial and final conditions are necessarily feasible. For notational convenience, let the equality constraints (14), (22), and (25) be represented by $F_x(\mathcal{X}) = 0$. Also, let equality constraints (15), (23), and (26) be represented by $F_y(\mathcal{Y}) = 0$. Similarly, let $C_x(\mathcal{X}) \le 0$ denotes the inequality constraints (16) and (18), and let $C_y(\mathcal{Y}) \le 0$ denotes the inequality constraints (17) and (19). Then the energy-optimal motion planning problem can be formulated as the following compact form

Problem 1 (Energy-optimal motion planning):

$$\begin{aligned}\min \quad & \tilde{J}(\mathcal{X}, \mathcal{Y}) \\ \text{subject to} \quad & F_x(\mathcal{X}) = 0, \quad C_x(\mathcal{X}) \le 0, & (28) \\ & F_y(\mathcal{Y}) = 0, \quad C_y(\mathcal{Y}) \le 0, & (29) \\ & D(\mathcal{X}, \mathcal{Y}) \in \mathcal{D}^{N-1}, & (30)\end{aligned}$$

where $D(\mathcal{X}, \mathcal{Y}) \in \mathcal{D}^{N-1}$ denotes the collision-avoidance constraint (27) for nodes with indices $1, \ldots, N-1$.

E. A Decoupling Method for Collision Avoidance Constraint

Problem 1 can be easily solved using numerical optimization when $\mathcal{D}$ is convex, since the cost function is convex, and the other constraints (28) and (29) are linear. Problem 1 is difficult only when $\mathcal{D}$ is non-convex, and the collision-avoidance constraint is active in the optimal solution. Next we propose a method to address this difficulty by taking advantage of the decoupled constraints and cost function in Problem 1.

Suppose that $\mathcal{X}_c$ and $\mathcal{Y}_c$ satisfy $D(\mathcal{X}_c, \mathcal{Y}_c) \in \mathcal{D}$. Because the motions in x and y directions are decoupled, we may fix the motion in y, and plan the motion in the x direction to minimize the cost function (11) while satisfying constraints including the collision avoidance constraint.

For example, consider the motion of a system which starts from (x_0, y_0) and moves to (x_f, y_f). As shown in Fig. 1, the trajectory of the system is represented by the dotted curve connecting the initial and final positions marked by blue dots. The green circles on the trajectory correspond to the position of the system at each time instance in $\{t_i\}_{i=1}^N$. It is clear from the figure that $\mathcal{D}$ is non-convex. By fixing the motion of the

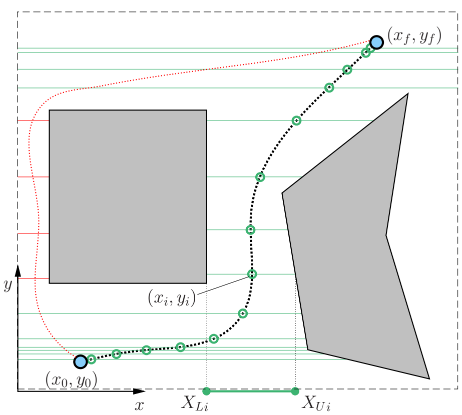

Fig. 1: Fixed motion in the y direction.

system in the y direction, the green circles can only move in the x direction, and the range of the movement is determined by $D(\mathcal{X}, \mathcal{Y}_c) \subset \mathcal{D}^{N-1}$ while ensuring that $\mathcal{X}_c$ is in this range. Specifically, let (x_{c_i}, y_{c_i}) be the node of the current motion $(\mathcal{X}_c, \mathcal{Y}_c)$ corresponding to time instance t_i, then the collision avoidance is ensured at node i as long as $x_i \in [X_{L_i}, X_{U_i}]$, where $[X_{L_i}, X_{U_i}]$ is the largest interval such that $[X_{L_i}, X_{U_i}] \times y_{c_i} \subset \mathcal{D}$ and $x_{c_i} \in [X_{L_i}, X_{U_i}]$. Let $D_x : \mathbb{R}^{2N+1} \to \mathbb{R}^{N-1}$ denotes the function mapping from $\mathcal{X}$ to the vector of x coordinates of nodes with indices $1, \dots, N-1$, which is given by the left-hand-side of (27), and let $X_L = [X_{L_1}, \dots, X_{L_{N-1}}]^T$, $X_U = [X_{U_1}, \dots, X_{U_{N-1}}]^T$. Since the motion in the y direction is fixed, the energy consumption $\tilde{J}(\mathcal{Y}_c)$ remains constant. Therefore the best improvement of energy efficiency with fixed y-motion is obtained by solving the following convex sub-problem

Problem 2 (Energy-optimal motion in the x direction):

$$\begin{aligned} \min \quad & \tilde{J}_x(\mathcal{X}) \\ \text{subject to} \quad & F_x(\mathcal{X}) = 0, \quad C_x(\mathcal{X}) \leq 0, \\ & X_L \leq D_x(\mathcal{X}; x_0) \leq X_U, \end{aligned}$$

Similarly, by fixing the motion in the x direction, we can plan the motion in the y direction by solving the following convex sub-problem

Problem 3 (Energy-optimal motion in the y direction):

$$\begin{aligned} \min \quad & \tilde{J}_y(\mathcal{Y}) \\ \text{subject to} \quad & F_y(\mathcal{Y}) = 0, \quad C_y(\mathcal{Y}) \leq 0, \\ & Y_L \leq D_y(\mathcal{Y}; y_0) \leq Y_U, \end{aligned}$$

Therefore, by solving Problem 2 and Problem 3 alternatively, the optimality of the trajectory can be improved while satisfying all constraints.

As shown in Fig. 1, when the red dotted line is used as the initial trajectory, the planning result using this method would certainly be different from the case when the optimization is started from the black dotted line, i.e., the solution of this method depends on the initial trajectory. However, given the convexity of the sub-problems, this method is both reliable and efficient. This is appealing for practical applications as compared to the exhaustive search procedure required by the MILP/MIQP method. As long as a reasonably good initial trajectory is used, the proposed method can further improve the initial trajectory for better energy performance with the guarantee of feasibility. Because both the path geometry and the speed along the path are planned simultaneously, this method can also achieve better optimality than the two-layer decomposition approach introduced in Section I.

One limitation of the proposed method is that it requires a trajectory to get started. The starting trajectory is used only for determining the collision-avoidance constraint for solving the first subproblem. For many systems such as robotic manipulators, automobiles and aircraft, it is easy to generate such a trajectory, which can even be made feasible for most of the time. The two-layer decomposition approach can also be applied to obtain the initial trajectory. For more details about initial trajectory generation techniques, interested reader may refer to Refs. [11], [4], [1].

III. A Quadratic Programming Energy-Efficient Motion Planning Algorithm

A. Quadratic Program Matrices

As shown in (11), the cost functions $\tilde{J}_x(\mathcal{X})$ and $\tilde{J}_y(\mathcal{Y})$ are quadratic in $\mathcal{X}$ and $\mathcal{Y}$, respectively, and the constraints in Problem 2 and Problem 3 are linear. Therefore, Problem 2 and Problem 3 are quadratic programs, which can be solved efficiently using standard QP solvers. In particular, Problem 2 is formulated as below:

$$\begin{aligned} \min \quad & \tilde{J}_x = \mathcal{X}^T \mathbf{H_x} \mathcal{X} \\ \text{subject to} \quad & \mathbf{A}\mathcal{X} \leq \mathbf{b_x}, \quad \mathbf{E_x}\mathcal{X} = \mathbf{d_x}, \end{aligned} \tag{31}$$

where

$$\mathbf{H_x} = \left[\begin{array}{c|c} \frac{1}{2} Q_x \mathbf{H_{xv}} & \mathbf{0} \\ \hline \mathbf{0} & R_x \mathbf{\Delta} \end{array} \right],$$

and $\mathbf{H_{xv}} = \mathrm{diag}([\Delta_1, \Delta_1 + \Delta_2, \dots, \Delta_{N-1} + \Delta_N, \Delta_N])$, $\mathbf{\Delta} = \mathrm{diag}([\Delta_1, \dots, \Delta_N])$.

$$\mathbf{A} = \left[\begin{array}{cc} \mathbf{I}_{N+1} & 0 \\ -\mathbf{I}_{N+1} & 0 \\ 0 & \mathbf{I}_N \\ 0 & -\mathbf{I}_N \\ \hline \multicolumn{2}{c}{\mathbf{M}_1} \\ \multicolumn{2}{c}{\vdots} \\ \multicolumn{2}{c}{\mathbf{M}_{N-1}} \\ \multicolumn{2}{c}{-\mathbf{M}_1} \\ \multicolumn{2}{c}{\vdots} \\ \multicolumn{2}{c}{-\mathbf{M}_{N-1}} \end{array} \right], \quad \mathbf{b_x} = \left[\begin{array}{c} v_{x_{\max}} \mathbf{1}^{(N+1)\times 1} \\ -v_{x_{\min}} \mathbf{1}^{(N+1)\times 1} \\ u_{x_{\max}} \mathbf{1}^{N\times 1} \\ -u_{x_{\min}} \mathbf{1}^{N\times 1} \\ X_U - x_0 \\ -X_L + x_0 \end{array} \right],$$

where $\mathbf{M}_k = \frac{1}{2}[\Delta_1, \Delta_1 + \Delta_2, \dots, \Delta_{k-1} + \Delta_k, \Delta_k, \mathbf{0}^{1\times(2N-k)}]$ for $k = 1, \dots, N$. $\mathbf{E_x} = [\mathbf{E_{x1}}^T, \mathbf{E_{x2}}^T, \mathbf{M}_N^T]^T$, $\mathbf{d_x} = [\mathbf{0}^{1\times N}, v_{x_0}, v_{x_f}, x_f - x_0]^T$, and

$$\mathbf{E_{x1}} = \left[\begin{array}{cccc|c} p_{x_i} + 2 & p_{x_i} - 2 & \cdots & 0 & \\ \vdots & \ddots & \ddots & \vdots & 2b_x \mathbf{\Delta} \\ 0 & \cdots & p_{x_i} + 2 & p_{x_i} - 2 & \end{array} \right],$$

$$\mathbf{E_{x2}} = \left[\begin{array}{ccc|c} 1 & \cdots & 0 & \mathbf{0}^{2\times N} \\ 0 & \cdots & 1 & \end{array} \right],$$

where $p_{x_i} = d_x \Delta_i$. Similarly, Problem 3 can also be formulated as a QP problem:

$$\begin{aligned} \min \quad & \tilde{J}_y = \mathcal{Y}^T \mathbf{H}_\mathbf{y} \mathcal{Y} \\ \text{subject to} \quad & \mathbf{A}\mathcal{Y} \leq \mathbf{b}_\mathbf{y}, \quad \mathbf{E}_\mathbf{y}\mathcal{Y} = \mathbf{d}_\mathbf{y}. \end{aligned} \tag{32}$$

The matrices in (32) are similar to those in (31), hence are omitted for brevity.

B. An Energy-Efficient Collision Avoidance QP Algorithm

we propose the following algorithm for solving Problem 1 using the method introduced in the previous section:

Algorithm 1 (Alternating Quadratic Programming Trajectory Optimization)

1) Choose a time grid $\{t_i\}_{i=0}^N$, and obtain a set of $\mathcal{X}_0$ and $\mathcal{Y}_0$ such that $D(\mathcal{X}_0, \mathcal{Y}_0) \subset \mathcal{D}^{N-1}$. Let $i = 0$, and $\mathcal{Y}_c = \mathcal{Y}_0$.
2) Let $i = i + 1$. Determine X_L and $X_U \in \mathbb{R}^{(N-1)}$ such that $\mathcal{B}_x = \{\mathcal{X} | X_L \leq \mathcal{X} \leq X_U\}$ is the largest set satisfying
   a) $\mathcal{B}_x \ni \mathcal{X}_{i-1}$
   b) $D(\mathcal{X}, \mathcal{Y}_{i-1}) \subset \mathcal{D}^{(N-1)}$ for any $\mathcal{X} \in \mathcal{B}_x$

   Solve the QP problem (31) for $\mathcal{X}^*$. Let $\mathcal{X}_i = \mathcal{X}^*$, and $\mathcal{X}_c = \mathcal{X}^*$.
3) Determine Y_L and $Y_U \in \mathbb{R}^{(N-1)}$ such that $\mathcal{B}_y = \{\mathcal{Y} | Y_L \leq \mathcal{Y} \leq Y_U\}$ is the largest set satisfying
   a) $\mathcal{B}_y \ni \mathcal{Y}_{i-1}$,
   b) $D(\mathcal{X}_c, \mathcal{Y}) \subset \mathcal{D}^{(N-1)}$ for any $\mathcal{Y} \in \mathcal{B}_y$

   Solve the QP problem (32) for $\mathcal{Y}^*$. Let $\mathcal{Y}_i = \mathcal{Y}^*$, and $\mathcal{Y}_c = \mathcal{Y}^*$.
4) Repeat steps 2 and 3 until the solution converges. Retrieve the state and control time histories from $\mathcal{X}^*$ and $\mathcal{Y}^*$.

Remark 3.1: The initial guesses $\mathcal{X}_0$ and $\mathcal{Y}_0$ chosen in step 1) of Algorithm 1 are used to establish the bounds X_L and X_U in the second step, and they are required to satisfy constraint (30) but not necessarily (28) and (29), because the QP solver is reliably find the feasible solution satisfying (28) and (29) when such a solution exists. However, as mentioned previously, a good initial guess can facilitate the convergence of the algorithm. Hence it is recommended to start the AQP algorithm from a feasible trajectory.

C. Convergence and Optimality

Throughout this section, we assume that the QP solver used in Algorithm 1 always returns the optimal (necessarily feasible) solution when such a solution exists, which is a valid assumption given the mature theory and solvers for Quadratic Programming. The following two propositions show that the AQP algorithm retains the feasibility of the solution as long as it is started properly, and monotonically improves the optimality.

Proposition 1: Let $\mathcal{X}_i$ and $\mathcal{Y}_i$ be the x and y motion planning results given by the i^{th} iteration of Algorithm 1. Suppose $(\mathcal{X}_0, \mathcal{Y}_0)$ is feasible for Problem 1, i.e., $(\mathcal{X}_0, \mathcal{Y}_0)$ satisfy all constraints in Problem 1, then $(\mathcal{X}_i, \mathcal{Y}_i)$ are feasible for all $i \geq 1$.

Proposition 2: Let $\mathcal{X}_i$ and $\mathcal{Y}_i$ be the x and y motion planning results by the i^{th} iteration of Algorithm 1, then $\tilde{J}(\mathcal{X}_i, \mathcal{Y}_i)$ decreases monotonically as i increases. Furthermore, the sequence $\{\tilde{J}(\mathcal{X}_i, \mathcal{Y}_i)\}$ converges as $i \to \infty$.

In general, the convergence of the cost function as shown in Proposition 2 does not guarantee the convergence of solution $(\mathcal{X}_i, \mathcal{Y}_i)$. However, due to the strict convexity of the cost function and the decoupled dynamics considered in this paper, Algorithm 1 indeed converges as shown by the following theorem.

Theorem 3.1: The sequence $(\mathcal{X}_i, \mathcal{Y}_i)$ as given by Algorithm 1 solving Problem 1 converges as $i \to \infty$.

Theorem 3.2: Suppose Algorithm 1 converges to a feasible solution $(\mathcal{X}^*, \mathcal{Y}^*)$. Assuming that the boundary of the collision-free region $\mathcal{D}$ is composed of piecewise linear segments which are parallel to either the x or the y axis. Then $(\mathcal{X}^*, \mathcal{Y}^*)$ is a locally optimal solution to Problem 1.

The proofs of results in this section are omitted. According to Theorem 3.2, the proposed AQP algorithm generates an optimal solution when the boundary of the collision-free region is a concatenation of zigzag lines parallel to the x or y axis. In the 3-dimensional case, the boundary should be compose of pieces of rectangles perpendicular to the x, y, and z axes to ensure the local optimality of the solution. For a general collision-free region, the AQP algorithm always converges, and the solution is not necessarily locally optimal. However, we can approximate the boundary of the collision-free region using zigzag lines and obtain a locally optimal solution for a problem with almost the same feasible region, hence achieve good optimality for the original problem.

IV. Examples

In this section, we use two numerical examples to demonstrate the proposed AQP algorithm. We limit the motion of the system in tunnels composed of connected squares, as shown in Fig. 2 and Fig. 3. In these two examples, the speed constraints $-160 \leq v_x, v_y \leq 160$ and control constraints $-5 \leq u_x, u_y \leq 5$ are enforced. The red dashed lines in these figures are the inial trajectories which are used to start Algorithm 1. These initial trajectories are generated by first calculating the minimum time trajectories connecting the corner points marked by red stars in these figures with zero inial and final speed, then relaxing the travel time by a certain factor greater than 1 to help with energy-saving. The energy-efficient trajectory generated by AQP algorithms are shown as blue curves in these figures. The energy-efficient trajectories are much smoother than the initial trajectories, and satisfy the collision avoidance constraint.

The speed and control profiles for the trajectory shown in Fig. 3 are given in Figs. 4 and 5, respectively. The speed and control constraints are clearly satisfied by the results.

Table I compares the energy consumption of trajectories shown in Figs. 2 and 3. The proposed energy-efficient motion planing algorithm is capable of substantially reducing the energy consumptions of motions in both the x and y directions. In order to better ensure collision avoidance between grid points, a grid refinement scheme is also implemented,

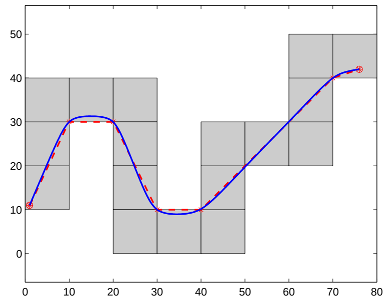

Fig. 2: Energy-efficient trajectory inside a tunnel, case 1.

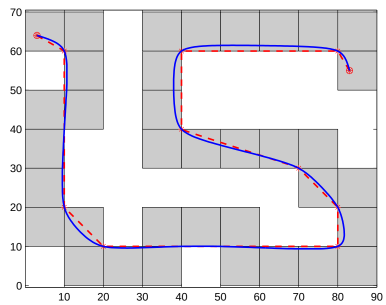

Fig. 3: Energy-efficient trajectory inside a tunnel, case 2.

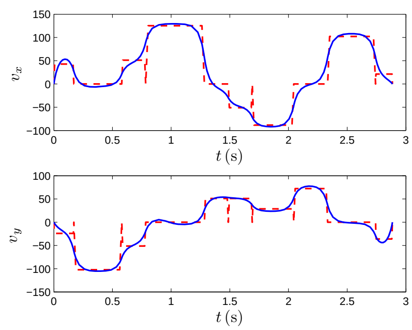

Fig. 4: Speed profile comparison, case 2.

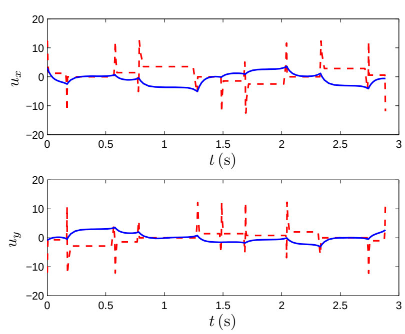

Fig. 5: Control profile comparison, case 2.

TABLE I: Energy consumption comparison.

		J_x	J_y	J	Energy saving	T_{CPU}(s)
case 1	heuristic	237.8	138.1	375.9	–	–
	AQP	190.0	92.7	282.7	24.8%	1.25
case 2	heuristic	73.1	76.1	149.2	–	–
	AQP	42.4	53.6	96.0	35.6%	0.36

which adaptively refines the time grid when collision occurs between neighboring grid points. Details about the grid refinement process are omitted in this paper due to limited space. The computation time depends on the grid size N. For case 1, $N = 211$ when the algorithm terminates. For the second case, the final grid size is $N = 120$. For both cases the AQP algorithm stops after four iterations including three local grid refinements.

V. Conclusion

In this paper we propose an Alternating Quadratic Programming (AQP) method for the energy-efficient collision avoidance trajectory generation of systems with decoupled linear dynamics. This method is highly efficient and reliable, hence, is suitable for real-time motion planning applications. The method can also be extended to cooperative collision-free trajectory planning involving multiple agents.

VI. Acknowledgement

The work in this paper was done at MERL.

References

[1] E. Bakolas, Y. Zhao, and P. Tsiotras, "Initial guess generation for aircraft landing trajectory optimization," in *AIAA Guidance, Navigation, and Control Conference*, no. AIAA-2011-6689, August 8-11 2011, pp. Portlanta, OR.

[2] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.

[3] O. Brock and L. E. Kavraki, "Decomposition-based motion planning: a framework for real-time motion planning in high-dimensional configuration spaces," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, May 2001, pp. 1469 – 1474.

[4] P. F. Gath, K. H. Well, and K. Mehlem, "Initial guess generation for rocket ascent trajectory," *AIAA Journal of Spacecraft and Rockets*, vol. 29, no. 4, pp. 515–521, Jul.-Aug. 2002.

[5] M. Jacobson and U. T. Ringertz, "Airspace constraints in aircraft emission trajectory optimization," *Journal of Aircraft*, vol. 47, no. 4, pp. 1256–1265, Jul.-Aug. 2010.

[6] J. J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue, "Dynamically-stablemotion planning for humanoid robots," *Autonomous Robots*, vol. 12, no. 1, pp. 105–118, Jan. 2002.

[7] D. Mellinger, A. Kushleyev, and V. Kumar, "Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams," in *IEEE International Conference on Robotics and Automation*, St. Paul, MN, May 14-18 2012, pp. 477 – 483.

[8] A. Richards, T. Schouwenaars, J. P. How, and E. Feron, "Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 4, pp. 755–763, July-August 2002.

[9] C. Shih, T. Lee, and W. Gruver, "Motion planning with time-varying polyhedral obstacles based on graph search and mathematical programming," in *In Proceeding of the IEEE International Conference on Robotics and Automation*, vol. 1, Cincinnatti, OH, USA, May 1990, pp. 331 – 337.

[10] Z. Shiller, "Time-energy optmal control of articulated systems with geometric path constraints," in *Proceeding of IEEE International Conference on Robotic and Automation*, San Diego, CA, 1994, pp. 2680–2685.

[11] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Transactions on Automatic Control*, vol. AC-30, no. 6, pp. 531–541, Jun. 1985.

[12] D. Verscheure, B. Demeulenaere, J. Swevers, J. D. Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Transaction on Automatic Control*, vol. 54, no. 10, pp. 2318–2327, Oct. 2009.

[13] Y. Zhao and P. Tsiotras, "Analysis of energy-optimal aircraft landing operation trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 3, pp. 833–845, 2013.