

Energy-Optimal Collision-Free Motion Planning for Multi-Axis Motion Systems: An Alternating Quadratic Programming Approach

Zhao, Y.; Wang, Y.; Zhou, M.; Wu, J.

TR2018-160 December 07, 2018

Abstract

This work investigates energy-optimal motion planning for a class of multi-axis motion systems where the system dynamics are linear time-invariant and decoupled in each axis. Solving the problem in a reliable and efficient manner remains challenging owing to the presence of various constraints on control and state, non-convexity in the cost function, and obstacles. This work shows how the cost function can be convexified by taking into account the system dynamics, while decomposing decision variables to obtain a convex representation of collision avoidance constraints. With the convexified cost function and constraints, the original problem is decomposed into two quadratic programming (QP) problems. An alternating quadratic programming (AQP) algorithm is proposed to solve both QP problems alternately and iteratively until convergence. Requiring an initial feasible trajectory as a guess, AQP necessarily converges to an energy-efficient solution which is homotopic to the initial guess. Under certain circumstances, AQP is guaranteed to produce a local optimum. Simulation demonstrates that AQP is computationally efficient and reliable while claiming comparable energy saving as the Mixed-Integer QP approach.

IEEE Transactions on Automation Science and Engineering

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Energy-Optimal Collision-Free Motion Planning for Multi-Axis Motion Systems: An Alternating Quadratic Programming Approach

Yiming Zhao, *Member, IEEE*, Yebin Wang, *Senior Member, IEEE*, MengChu Zhou, *Fellow, IEEE*, and Jing Wu, *Member, IEEE*

Abstract—This work investigates energy-optimal motion planning for a class of multi-axis motion systems where the system dynamics are linear time-invariant and decoupled in each axis. Solving the problem in a reliable and efficient manner remains challenging owing to the presence of various constraints on control and state, non-convexity in the cost function, and obstacles. This work shows how the cost function can be convexified by taking into account the system dynamics, while decomposing decision variables to obtain a convex representation of collision avoidance constraints. With the convexified cost function and constraints, the original problem is decomposed into two quadratic programming (QP) problems. An alternating quadratic programming (AQP) algorithm is proposed to solve both QP problems alternately and iteratively until convergence. Requiring an initial feasible trajectory as a guess, AQP necessarily converges to an energy-efficient solution which is homotopic to the initial guess. Under certain circumstances, AQP is guaranteed to produce a local optimum. Simulation demonstrates that AQP is computationally efficient and reliable while claiming comparable energy saving as the Mixed-Integer QP approach.

Notes to Practitioners—This paper presents an energy-optimal motion planning algorithm that can be easily implemented on a class of multi-axis motion systems. Main advantages of the proposed algorithm are: 1) it produces a trajectory resulting in lower but comparable energy efficiency as the global optimum; 2) it is guaranteed to provide an energy-efficient and constraint-compliant trajectory, and thus is reliable; 3) it requires a low computational load and can be deployed on a wide range of applications; and 4) its implementation is straightforward to any engineer with basic knowledge of numerical methods.

Index Terms—Motion planning, minimum energy, obstacle avoidance, quadratic programming, constrained optimization

I. INTRODUCTION

THIS paper focuses on energy-optimal collision-free motion planning for multi-axis motion systems with linear time-invariant (LTI) dynamics. The dynamics in each axis are decoupled from the rest axes, whereas the motion is coupled with other axes through non-convex collision avoidance constraints and a cost function. The system motion is additionally subject to convex constraints such as speed, acceleration, and

control constraints. Examples of such systems include multi-stage positioning machines, chip mounting machines, and drilling machines.

The energy-optimal collision-free motion planning can be found in extensive applications, to name a few, machines, mobile robots, and manipulators. Prevalent approaches to tackle this problem can be categorized into decomposition-based [1] and kinodynamic approaches [2]. For the decomposition approach, a path planner takes into account geometric conditions of the environment and robot to generate a collision-free and energy-efficient path [1], [3], [4]; and a motion planner determines a dynamically feasible and energy-efficient motion along the path. The energy-efficient path planning usually resorts to graph-based search algorithms, e.g. A* and its variants [5], [6]. The graph is typically constructed by connecting collision-free grids, lattices, or random nodes, which represent collision-free configurations of the robot. The grids are obtained by decomposing the configuration space, whereas lattices and random nodes can be generated by sampling the configuration space or control input space. Well-established sampling-based algorithms for graph construction include probabilistic roadmap [7], rapid-exploring random tree (RRT) [8], etc. Interested readers are referred to [9] for more information. The motion along a given path can be determined by applying, for instance, the minimum principle [10], [11], dynamic programming [12], and numerical optimization [13]–[15]. The minimum principle is merely applicable to low order systems with simple constraints, whereas numerical optimization can deal with more general systems and constraints. For the numerical optimization approach, energy-efficient motion planning for a given path ends up with solving a non-convex optimization problem, which suffers a high computational load and lack of convergence guarantee. Compared with kinodynamic planning, the decomposition approach is relatively computationally efficient, albeit plagued by sub-optimality. Meanwhile, since the geometric path may not satisfy the system dynamics, the dynamically feasible motion might deviate from it and collide with obstacles [3]. This limitation is apparent when the system dynamics are nonholonomic. In order to avoid non-convex collision avoidance constraints in the motion planning stage, existing work largely focus on the determination of the velocity profile and thus achieve computational efficiency; however little work exploits the freedom of finely tuning the geometric path to improve energy efficiency.

Y. Zhao is with Halliburton Energy Service, 3000 N Sam Houston Pkwy E, Houston, TX 77032, USA. email: yiming.zhao@ieee.org

Y. Wang is with Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA 02139, USA. email: yebinwang@ieee.org

M. C. Zhou is with Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA. email: zhou@njit.edu

J. Wu is with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China. email: jingwu@sjtu.edu.cn

Determining the path and motion simultaneously, kinodynamic planning addresses limitations of the decomposition approach at the expenses of a higher computational cost. A core question underlying kinodynamic planning is how to construct a dynamically feasible path between any two nodes in the configuration or system state space. This is non-trivial for systems with nonlinear dynamics or numerous constraints, because it corresponds to solve a boundary value problem. This hurdle can be effectively overcome by sampling the control input space, by which the construction of the dynamically feasible path boils down to solving an initial value problem [16]. Such a treatment however incurs the loss of algorithmic completeness [17]. Alternatively, many studies exploit the numerical optimization approach [18], [19], which is usually more tractable than those based on classical optimal control theory, e.g. [20]. Owing to non-convexity in dynamics and constraints, this approach results in solving non-convex optimization problems, which rarely has guaranteed convergence and optimality. As a common practice, integer variables are introduced such that for fixed values of the integer variables, collision avoidance constraints are greatly simplified or even convex. Such a scheme yields, for instance, a Mixed-Integer Linear Programming (MILP) [21] and a Mixed-Integer Quadratic Programming (MIQP) [22]. Although an MILP or MIQP problem can be solved for the globally optimal solution, its computation time grows exponentially along with the number of obstacles and the computation becomes prohibitive.

It is worth mentioning that time-optimal motion planning has been intensively studied via similar approaches. Work [23], [24] considers time-optimal velocity planning along a given path by utilizing optimal control theory; [25] follows the numerical optimization approach and demonstrates that time-optimal motion planning for a wide range of mechanical systems can be formulated as a convex programming problem. Key limitation of work [25] is two-fold: the mechanical systems should not include viscous friction; the cost function has to be either time, or copper loss, or their combination. Work [2] performs kinodynamic time-optimal motion planning by constructing and searching a directed graph which approximates the state space of the robot. Given a finite set of control actions and the robot's initial state, the directed graph is recursively constructed by applying control actions for a fixed time interval to the robot dynamics; and then the breadth first search is conducted until a path, rooted in the initial state, extends to a state close enough to the goal state. Having the directed graph induced by a finite set of control actions is plausible for the time-optimal problem, because the optimal control admits bang-bang actions exactly captured by the finite set. With this fact, the kinodynamic planning converges to the optimal solution as the time interval goes to zero. Nevertheless a fixed time interval results in a finite discretization of the state space consisting of position and velocity state variables; and the resultant motion is not only sub-optimal, but also ϵ -far from the goal state [2]. Turning to the energy-optimal problem, its optimal control is continuous. Although the idea in [2] can be generalized to the energy-optimal problem, one cannot establish the asymptotic optimality even with an infinitesimal

time interval. This is consistent with observation in [26] where the asymptotic optimality of the RRT* algorithm is established under the condition of optimal steering. Additionally, kinodynamic planning by assuming a finite set of control actions necessarily leads to motions with discontinuous control, which is undesirable for possible excitation of vibration. Hence, it is advantageous to smooth such motions in a post-processing stage.

This work exploits the numerical optimization approach to solve the energy-optimal motion planning problem and mainly addresses two challenges: lack of convergence guarantee and proof of local optimality. With an initial feasible trajectory being a prerequisite, this work concentrates on the motion planning stage of the decomposition approach. An alternating programming approach is followed to utilize the design freedom in path as well as velocity, and thus ameliorate energy efficiency of the resultant solution. Main contributions of this work are: 1) show how a cost function characterizing the copper losses and mechanical work of electric motors can be convexified by taking into account the system dynamics; 2) offer an Alternating Programming approach to refine a dynamically feasible trajectory to improve energy efficiency; 3) identify a class of energy-optimal motion planning problems, for which the convergence of the proposed alternating programming algorithm can be established, and the resultant solution is locally optimal. More specifically, this work first convexifies the cost function, by which the motion planning problem admits MIQP formulation. Additionally, we circumvent the non-convexity of collision avoidance constraints by decomposing decision variables into multiple sets, where each set corresponds to one individual axis. Taking advantage of the decoupled nature of system dynamics, constraints, and the cost function, we decompose the motion planning problem into multiple Quadratic Programming (QP) subproblems. An Alternating Quadratic Programming (AQP) algorithm is proposed to perform motion planning by solving subproblems alternatively and iteratively.

As a continuation and completion of [27], this work establishes the convergence of the AQP algorithm. Furthermore, when the boundary of a non-convex collision-free region satisfies certain conditions, AQP always yields a locally optimal solution. Comparing AQP and MIQP through simulation, AQP can determine a solution resulting comparable energy efficiency as the solution of MIQP but require much less computation time. It is noteworthy that AQP can be viewed as a special case of alternating convex programming, which has been applied to medical image processing [28] and joint optimization of communication and control systems [29]. This work is related to [10], where a single-axis motion system moves in an obstacle-free environment. Requiring a dynamically feasible trajectory as its input, the proposed alternating programming module can be cascaded with existing work, e.g. [2], to further improve energy efficiency.

This paper is structured as follows. Section II introduces the system and energy-optimal collision-free motion planning problems. Section III presents the convexification of a cost function, decomposition of the motion planning problem, and AQP. The convergence of AQP and optimality of its solution

are analyzed in Section IV. Numerical validation are provided in Section V. Section VI concludes this paper.

II. PROBLEM FORMULATION

This paper presents results based on a multi-axis motion system having two translational degree of freedoms (DoFs). This causes no loss of generality because the proposed work can be easily extended to systems with more DoFs.

A. System Dynamics

Assume that a two-DoF multi-axis motion system has the following LTI dynamics

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

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

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

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

where $x, y \in \mathbb{R}$ denote the displacements in the x and y axis, respectively; $v_x, v_y \in \mathbb{R}$ the speeds along the x and y axis, respectively; u_x, u_y are the control inputs; $d_x, d_y > 0$ are viscous friction coefficients; and $b_x, b_y > 0$ are constants. Note that parameters in (3) and (4) are normalized by system masses, and the motion system can have distinct masses in different axes. The system starts from an initial position (x_0, y_0) at time $t = 0$ with initial speeds (v_{x_0}, v_{y_0}) , and moves to a final position (x_f, y_f) at time $t = t_f$ with final speeds (v_{x_f}, v_{y_f}) . The nonlinear Coulomb friction effect is neglected in exchange for improved computation efficiency and guaranteed convergence of the algorithm later proposed. This simplification is adopted in literature, e.g. [30]–[32], and plausible for high performance machines where the Coulomb force is typically much smaller than actuator forces.

B. Continuous-Time Problem

The energy consumption of system (1)-(4) is characterized by the following quadratic cost function

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

where $R_x u_x^2$ and $R_y u_y^2$ correspond to copper losses, and $K_x v_x u_x$ and $K_y v_y u_y$ mechanical work. Constants $R_x, R_y, K_x,$ and K_y are positive. Similar cost functions have been used in the literature to approximate the energy consumption of servo systems [10], [11].

Representation of obstacles is an important aspect of a motion planning problem formulation. Regular shapes such as polytopes, cylinders, and spheres are often used to describe obstacles [21], [33], [34], or alternatively, the feasible (obstacle-free) region or volume for system's motion [35]. This paper adopts the feasible region formulation, and assumes that a time-invariant obstacle-free region is represented by a closed compact set $\mathcal{D} \subset \mathbb{R}^2$. With the system being a point mass, \mathcal{D} can be readily represented by using boundaries of obstacles. Otherwise, one can inflate obstacles to account for the system geometry, and reduce the system to a point mass.

The inflation may cause loss of optimality or even feasibility, which is however out of the scope of this work. For simplicity, the system is assumed a point mass.

Problem 1 (Continuous-Time Energy-Optimal Motion Planning): Given the system (1)-(4), a final time t_f , the initial and final states

$$\begin{aligned} (x(0), v_x(0), y(0), v_y(0)) &= (x_0, v_{x_0}, y_0, v_{y_0}) \in \mathcal{D} \\ (x(t_f), v_x(t_f), y(t_f), v_y(t_f)) &= (x_f, v_{x_f}, y_f, v_{y_f}) \in \mathcal{D}, \end{aligned} \quad (6)$$

and an obstacle-free compact set $\mathcal{D} \subset \mathbb{R}^2$, find $u_x, u_y \in \mathcal{PC}[0, t_f]^1$ which minimize the cost function (5). While along the system trajectory, both the system state and control satisfy:

- 1) dynamic constraints (1) to (4),
- 2) boundary conditions (6),
- 3) speed constraints

$$\begin{aligned} v_{x_{\min}} \leq v_x(t) \leq v_{x_{\max}} \\ v_{y_{\min}} \leq v_y(t) \leq v_{y_{\max}}, \quad t \in [0, t_f], \end{aligned} \quad (7)$$

where $v_{x_{\min}}, v_{x_{\max}}, v_{y_{\min}}, v_{y_{\max}}$ are constants,

- 4) control constraints

$$\begin{aligned} u_{x_{\min}} \leq u_x(t) \leq u_{x_{\max}} \\ u_{y_{\min}} \leq u_y(t) \leq u_{y_{\max}}, \quad t \in [0, t_f], \end{aligned} \quad (8)$$

where $u_{x_{\min}}, u_{x_{\max}}, u_{y_{\min}}, u_{y_{\max}}$ are constants

- 5) collision-avoidance constraints

$$(x(t), y(t)) \in \mathcal{D}, \quad t \in [0, t_f]. \quad (9)$$

Without collision-avoidance constraints, feasibility of Problem 1 can be investigated by applying classic optimal control theory. On the contrary, one resorts to complete algorithms to detect feasibility [9]. For simplicity, we assume that Problem 1 is feasible, and tackle challenges arising from the non-convexity of the cost function (5) and the domain \mathcal{D} . In order to guarantee convergence of the proposed algorithm, we further impose the following restriction: the cost function, constraints, and the system dynamics can be decoupled for different translational motions. As a result, when collision avoidance constraints (9) are absent, Problem 1 can be equivalently solved by performing motion planning for the x and y subsystems separately. The motion planning problem for each subsystem (subproblem) can be solved efficiently by using numerical optimization. In the presence of obstacles, however, subproblems are coupled via collision avoidance constraints (9). In the following, a numerical optimization method is developed to efficiently handle such a case.

C. Discretized Problem

Both the cost function and system dynamics are discretized on a grid in the time domain by using the trapezoidal integration rule. Let the time grid (possibly non-uniform) be $\{t_i\}_{i=0}^N$ with $t_i \in [t_0, t_f]$ and N being the number of grids. Note $t_0 = 0$ and $t_N = t_f$. Also, let $\Delta_i = t_i - t_{i-1}$ for any $i = 1, \dots, N$. The state and control variables are discretized on the grid $\{t_i\}_{i=0}^N$ as follows: for $i = 0, \dots, N$,

$$x_i = x(t_i), \quad v_{x_i} = v_x(t_i), \quad y_i = y(t_i), \quad v_{y_i} = v_y(t_i),$$

¹ $\mathcal{PC}([0, t_f])$ is the functional space consisting of all piecewise-continuous functions defined over $[0, t_f]$.

and for $i = 1, \dots, N$,

$$u_{x_i} = u_x \left(\frac{t_i + t_{i-1}}{2} \right), \quad u_{y_i} = u_y \left(\frac{t_i + t_{i-1}}{2} \right).$$

For notational convenience, let

$$\begin{aligned} X &= [x_0, x_1, \dots, x_N]^T \\ V_x &= [v_{x_0}, v_{x_1}, \dots, v_{x_N}]^T \\ U_x &= [u_{x_1}, u_{x_2}, \dots, u_{x_N}]^T \\ Y &= [y_0, y_1, \dots, y_N]^T \\ V_y &= [v_{y_0}, v_{y_1}, \dots, v_{y_N}]^T \\ U_y &= [u_{y_1}, u_{y_2}, \dots, u_{y_N}]^T \\ \mathcal{X} &= [V_x^T, U_x^T]^T \\ \mathcal{Y} &= [V_y^T, U_y^T]^T. \end{aligned}$$

The cost function J is discretized as

$$\begin{aligned} J(\mathcal{X}, \mathcal{Y}) &= J_x(\mathcal{X}) + J_y(\mathcal{Y}) \\ &= \sum_{i=1}^N \Delta_i \left(R_x U_{x_i}^2 + \frac{K_x}{2} (V_{x_{i-1}} + V_{x_i}) U_{x_i} \right) \\ &\quad + \sum_{i=1}^N \Delta_i \left(R_y U_{y_i}^2 + \frac{K_y}{2} (V_{y_{i-1}} + V_{y_i}) U_{y_i} \right). \end{aligned} \quad (10)$$

System dynamics (1)-(4) are enforced between neighboring grid points by the following linear equations

$$X_i - X_{i-1} = \Delta_i (V_{x_i} + V_{x_{i-1}}) \quad (11)$$

$$Y_i - Y_{i-1} = \Delta_i (V_{y_i} + V_{y_{i-1}}) \quad (12)$$

$$\frac{V_{x_i} - V_{x_{i-1}}}{\Delta_i} = -\frac{d_x}{2} (V_{x_i} + V_{x_{i-1}}) + b_x U_{x_i} \quad (13)$$

$$\frac{V_{y_i} - V_{y_{i-1}}}{\Delta_i} = -\frac{d_y}{2} (V_{y_i} + V_{y_{i-1}}) + b_y U_{y_i}, \quad (14)$$

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

$$v_{x_{\min}} \mathbf{1}_{N+1} \leq V_x \leq v_{x_{\max}} \mathbf{1}_{N+1} \quad (15)$$

$$v_{y_{\min}} \mathbf{1}_{N+1} \leq V_y \leq v_{y_{\max}} \mathbf{1}_{N+1} \quad (16)$$

$$u_{x_{\min}} \mathbf{1}_N \leq U_x \leq u_{x_{\max}} \mathbf{1}_N \quad (17)$$

$$u_{y_{\min}} \mathbf{1}_N \leq U_y \leq u_{y_{\max}} \mathbf{1}_N, \quad (18)$$

where all elements of $\mathbf{1}_k \in \mathbb{R}^k$ are 1. The other constraints include initial and final conditions

$$X_1 = x_0, \quad X_{N+1} = x_f \quad (19)$$

$$Y_1 = y_0, \quad Y_{N+1} = y_f \quad (20)$$

$$V_{x_1} = v_{x_0}, \quad V_{x_N} = v_{x_f} \quad (21)$$

$$V_{y_1} = v_{y_0}, \quad V_{y_N} = v_{y_f}, \quad (22)$$

and collision-avoidance constraints

$$(X_i, Y_i) \in \mathcal{D}, \quad i = 1, \dots, N + 1. \quad (23)$$

In order to reduce the number of decision variables, we eliminate parameters X_i and Y_i from boundary conditions (19)-(20) and collision-avoidance constraints (23) using (11) and (12), and obtain the following equivalent conditions:

$$\frac{1}{2} \sum_{i=1}^N \Delta_i (V_{x_i} + V_{x_{i-1}}) = x_f - x_0 \quad (24)$$

$$\frac{1}{2} \sum_{i=1}^N \Delta_i (V_{y_i} + V_{y_{i-1}}) = y_f - y_0 \quad (25)$$

$$\begin{aligned} &\left(\frac{1}{2} \sum_{i=1}^k \Delta_i (V_{x_i} + V_{x_{i-1}}), \frac{1}{2} \sum_{i=1}^k \Delta_i (V_{y_i} + V_{y_{i-1}}) \right) \\ &+ (x_0, y_0) \in \mathcal{D}, \quad k = 1, \dots, N - 1. \end{aligned} \quad (26)$$

The collision avoidance constraints are not enforced at the first and the last nodes since the initial and final conditions are necessarily feasible. For notational convenience, let equality constraints (13), (21), and (24) be represented by $F_x(\mathcal{X}) = 0$; and let equality constraints (14), (22), and (25) be represented by $F_y(\mathcal{Y}) = 0$. Similarly, let $C_x(\mathcal{X}) \leq 0$ denote (15) and (17), and $C_y(\mathcal{Y}) \leq 0$ denote (16) and (18). Problem 1 is discretized into the following compact form.

Problem 2 (Numerical Energy-Optimal Motion Planning):

$$\min J(\mathcal{X}, \mathcal{Y})$$

$$\text{subject to } F_x(\mathcal{X}) = 0, \quad C_x(\mathcal{X}) \leq 0 \quad (27)$$

$$F_y(\mathcal{Y}) = 0, \quad C_y(\mathcal{Y}) \leq 0 \quad (28)$$

$$D(\mathcal{X}, \mathcal{Y}) \in \mathcal{D}^{N-1}, \quad (29)$$

where $D(\mathcal{X}, \mathcal{Y}) \in \mathcal{D}^{N-1}$ denotes the collision-avoidance constraint (26) for nodes with indices $1, \dots, N - 1$.

Remark 2.1: Both the cost function $J(\mathcal{X}, \mathcal{Y})$ given by (10) and constraints (29) are non-convex, whereas constraints (27) and (28) are linear.

III. CONVEXIFICATION AND AQP ALGORITHM

A. Convexification of a Cost Function

The convexity of the cost function and the feasible region plays a crucial role in the convergence proof and computation efficiency. The cost function (5) is non-convex in v_x , u_x , v_y and u_y . However, as shown below, insights can be obtained by analyzing the convexity of (5) on the manifold confined by system dynamics (3) and (4), such that Problem 1 can be solved with better computational efficiency and numerical reliability.

Multiplying both sides of (3) by v_x , we have:

$$v_x \dot{v}_x = -d_x v_x^2 + b_x u_x v_x. \quad (30)$$

Re-arranging (30) and integrating both sides on $[0, t_f]$ give

$$\begin{aligned} &\int_0^{t_f} K_x u_x v_x dt \\ &= \int_0^{t_f} \frac{K_x}{b_x} (v_x \dot{v}_x + d_x v_x^2) dt \\ &= \frac{K_x}{b_x} \left(\int_0^{t_f} v_x \dot{v}_x dt + \int_0^{t_f} d_x v_x^2 dt \right) \\ &= \frac{K_x}{b_x} \left(\int_0^{t_f} \frac{1}{2} dv_x^2 + \int_0^{t_f} d_x v_x^2 dt \right) \\ &= \frac{K_x}{b_x} \left(\frac{v_x^2(t_f) - v_x^2(0)}{2} + d_x \int_0^{t_f} v_x^2 dt \right) \\ &= \frac{d_x K_x}{b_x} \int_0^{t_f} v_x^2 dt + c_x, \end{aligned}$$

where c_x is a constant determined by the boundary conditions. Similarly, mechanical work due to a motion in y direction can be written as

$$\int_0^{t_f} K_y u_y v_y dt = \frac{d_y K_y}{b_y} \int_0^{t_f} v_y^2 dt + c_y.$$

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

$$\int_0^{t_f} (R_x u_x^2 + Q_x v_x^2 + R_y u_y^2 + Q_y v_y^2) dt + c_x + c_y.$$

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

$$\begin{aligned} \tilde{J}(v_x, u_x, v_y, u_y) &= \tilde{J}_x(v_x, u_x) + \tilde{J}_y(v_y, u_y) \\ &= \int_0^{t_f} (R_x u_x^2 + Q_x v_x^2 + R_y u_y^2 + Q_y v_y^2) dt, \end{aligned} \quad (31)$$

which is convex in v_x, u_x, v_y , and u_y . Definitions of \tilde{J}_x and \tilde{J}_y are obvious from (31). The cost function \tilde{J} is discretized by using the trapezoidal integration rule as

$$\begin{aligned} \tilde{J}(\mathcal{X}, \mathcal{Y}) &= \tilde{J}_x(\mathcal{X}) + \tilde{J}_y(\mathcal{Y}) \\ &= \sum_{i=1}^N \Delta_i \left(R_x U_{x_i}^2 + \frac{Q_x}{2} V_{x_{i-1}}^2 + \frac{Q_x}{2} V_{x_i}^2 \right) \\ &+ \sum_{i=1}^N \Delta_i \left(R_y U_{y_i}^2 + \frac{Q_y}{2} V_{y_{i-1}}^2 + \frac{Q_y}{2} V_{y_i}^2 \right). \end{aligned} \quad (32)$$

B. Decoupling of Collision Avoidance Constraints

With the convexified cost function (32), Problem 2 is still afflicted by the non-convexity of collision-avoidance constraints (29), which can be relieved by the decoupled structure of constraints and the cost function. Suppose that \mathcal{X}_c and \mathcal{Y}_c satisfy $D(\mathcal{X}_c, \mathcal{Y}_c) \in \mathcal{D}^{N-1}$. Because 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 (32) while ensuring constraints, or vice versa.

For example, consider a motion from (x_0, y_0) to (x_f, y_f) . As shown in Fig. 1, the system trajectory, projected onto the x - y plane, is represented by the black dotted curve. The green empty circles on the trajectory correspond to system positions at time instants $\{t_i\}_{i=1}^N$. It is clear that \mathcal{D} is non-convex. With the y -motion (represented by \mathcal{Y}_c) fixed, the green circles can only move in the x direction, and the range of the movement is determined by $D(\mathcal{X}, \mathcal{Y}_c) \subset \mathcal{D}^{N-1}$. Specifically, let (x_i, y_i) be the node of the current motion $(\mathcal{X}_c, \mathcal{Y}_c)$ corresponding to time instance t_i , then 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 $[X_{L_i}, X_{U_i}] \times y_i \subset \mathcal{D}$. Let $D_x : \mathbb{R}^{2N+1} \rightarrow \mathbb{R}^{N-1}$ denote 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 (26), and let $X_L = [X_{L_1}, \dots, X_{L_{N-1}}]^T$, and $X_U = [X_{U_1}, \dots, X_{U_{N-1}}]^T$. Since the motion in the y direction is fixed, the energy consumption $\tilde{J}(\mathcal{Y}_c)$ remains constant and constraints in y axis hold. Therefore the energy efficiency of the system motion with fixed y -motion can be improved by planning motion in the x direction, i.e., solving the following subproblem.

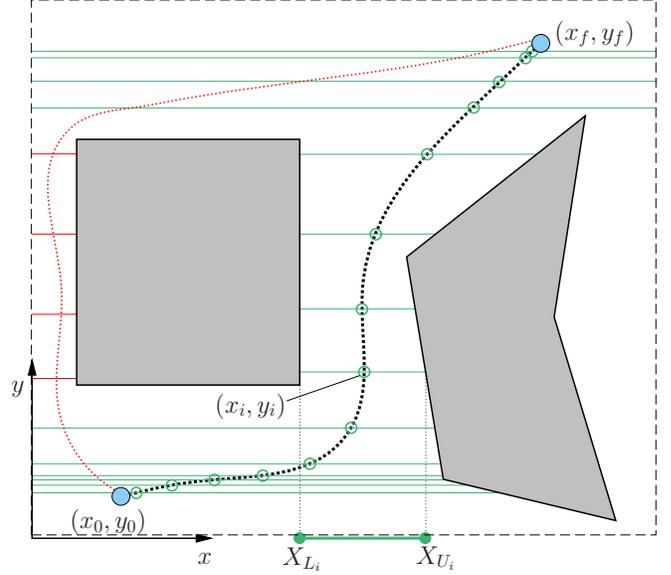


Fig. 1: Fixed motion in the y direction (gray region are obstacles)

Problem 3 (Energy-Optimal Motion Planning in x Axis):

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

By fixing the motion in x direction, the motion in y direction can be optimized by solving the following subproblem.

Problem 4 (Energy-Optimal Motion Planning in y Axis):

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

Both $D_x(\mathcal{X}; x_0)$ and $D_y(\mathcal{Y}; y_0)$ are linear mappings, and thus Problems 3-4 are convex.

C. Quadratic Program Matrices

As shown in (32), cost functions $\tilde{J}_x(\mathcal{X})$ and $\tilde{J}_y(\mathcal{Y})$ are quadratic in \mathcal{X} and \mathcal{Y} , respectively, and constraints in Problems 3-4 are linear. Therefore, Problems 3-4 are quadratic programs, which can be solved efficiently using standard QP solvers. In particular, Problem 3 is reformulated as below:

$$\begin{aligned} \min \quad & \tilde{J}_x = \mathcal{X}^T \mathbf{H}_x \mathcal{X} \\ \text{subject to} \quad & \mathbf{A} \mathcal{X} \leq \mathbf{b}_x, \quad \mathbf{E}_x \mathcal{X} = \mathbf{d}_x, \end{aligned} \quad (33)$$

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 $\mathbf{H}_{xv} = \text{diag}([\Delta_1, \Delta_1 + \Delta_2, \dots, \Delta_{N-1} + \Delta_N, \Delta_N])$, $\mathbf{\Delta} = \text{diag}([\Delta_1, \dots, \Delta_N])$. Matrices \mathbf{A} , \mathbf{b}_x , \mathbf{E}_x , and \mathbf{d}_x are given by

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_{N+1} & 0 \\ -\mathbf{I}_{N+1} & 0 \\ 0 & \mathbf{I}_N \\ 0 & -\mathbf{I}_N \\ \hline \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_{N-1} \\ -\mathbf{M}_1 \\ \vdots \\ -\mathbf{M}_{N-1} \end{bmatrix}, \quad \mathbf{b}_x = \begin{bmatrix} v_{x_{\max}} \mathbf{1}^{(N+1) \times 1} \\ -v_{x_{\min}} \mathbf{1}^{(N+1) \times 1} \\ u_{x_{\max}} \mathbf{1}^{N \times 1} \\ -u_{x_{\min}} \mathbf{1}^{N \times 1} \\ X_U - x_0 \\ -X_L + x_0 \end{bmatrix},$$

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

$$\mathbf{E}_{x1} = \left[\begin{array}{cccc|c} p_{x_1} + 2 & p_{x_1} - 2 & \cdots & 0 & 2b_x \Delta \\ \vdots & \ddots & \ddots & \vdots & \\ 0 & \cdots & p_{x_N} + 2 & p_{x_N} - 2 & \end{array} \right],$$

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

with $p_{x_i} = d_x \Delta_i$. Similarly, Problem 4 can also be formulated as a QP problem:

$$\begin{aligned} \min \quad & \tilde{J}_y = \mathcal{Y}^T \mathbf{H}_y \mathcal{Y} \\ \text{subject to} \quad & \mathbf{A}_y \mathcal{Y} \leq \mathbf{b}_y, \quad \mathbf{E}_y \mathcal{Y} = \mathbf{d}_y. \end{aligned} \quad (34)$$

The matrices in (34) are similar to those in (33), hence are omitted for brevity.

D. AQP Algorithm

By solving Problems 3-4 alternatively, the energy efficiency of the trajectory can be improved while all constraints hold. We have the following algorithm to solve Problem 2.

Algorithm 1 (AQP Algorithm)

- 1) Given a feasible solution $(x_0(t), y_0(t)), t \in [0, t_f]$ to Problem 1, choose a time grid $\{t_i\}_{i=0}^N$, and obtain a set of \mathcal{X}_0 and \mathcal{Y}_0 . 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) $\mathcal{X}_{i-1} \in \mathcal{B}_x$
 - b) $D(\mathcal{X}, \mathcal{Y}_{i-1}) \subset \mathcal{D}^{(N-1)}, \forall \mathcal{X} \in \mathcal{B}_x$.

Solve the QP problem (33) 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) $\mathcal{Y}_{i-1} \in \mathcal{B}_y$
 - b) $D(\mathcal{X}_c, \mathcal{Y}) \subset \mathcal{D}^{(N-1)}, \forall \mathcal{Y} \in \mathcal{B}_y$.

Solve the QP problem (34) 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 histories from \mathcal{X}^* and \mathcal{Y}^* .

Remark 3.1: Solving a dynamically feasible trajectory $(x_0(t), y_0(t))$ for Problem 1 is NP-hard. A plethora of work have been devoted to this problem and developed complete

or probabilistic complete planning algorithms, which return feasible solutions, if any, or failure [9]. This work focuses on a class of multi-axis motion systems, e.g. industrial machines, where the geometric layout of obstacles are static. Hence the search of a feasible path or trajectory could be greatly simplified and omitted here. Interested readers may refer to [36]–[38] for details.

Remark 3.2: The initial guess $(x_0(t), y_0(t))$ or $(\mathcal{X}_0, \mathcal{Y}_0)$ is used to establish bounds X_L and X_U in Step 2). Meanwhile, it has to be dynamically feasible and collision-free, i.e., satisfy constraints (27), (28) and (29), in order to guarantee convergence and feasibility of AQP. Numerical study demonstrates that when (27) and (28) are not satisfied by \mathcal{X}_0 and \mathcal{Y}_0 , the AQP may still recover feasibility and converge, albeit without guarantee.

A dynamically feasible and collision-free trajectory uniquely determines feasible domains \mathcal{D}_x and \mathcal{D}_y , which might be a subset of the original feasible region. Searching within the same homotopy class of solutions as the initial guess, Algorithm 1 may achieve global optimality, if the global optimal solution is homotopic to the initial guess. Algorithm 1 certainly does not yield the global optimal solution, if it is not homotopic to the initial guess. As shown in Fig. 1, when the red dotted line is used as an initial trajectory, the planning result would be different from the black dotted case. However, given the convexity of two subproblems, this method is appealing in terms of reliability and computational efficiency for practical applications as compared to solving a non-convex optimization problem blindly and the MIQP, respectively. As long as a good initial trajectory is used, the proposed method can improve energy efficiency with guaranteed feasibility.

E. Local Grid Refinement

Since solutions to Problem 2 are discrete, collision avoidance constraints are enforced only at a finite number of locations corresponding to the system position at each time instance $t_i \in \{t_i\}_{i=0}^N$. Inter-collision may occur between adjacent grid points, as shown in Fig. 2. Rather than introducing conservatism into the planning result by expanding the boundary of the obstacle, we exploit a grid refinement method to reduce inter-collision by increasing the time grid resolution around where inter-collision occurs. The local refinement approach clearly makes more efficient use of the discretization points than uniformly adding more grid points along the trajectory.

Specifically, between each pair of adjacent time grid points t_i and t_{i+1} , inter-collision is first detected by interpolating $(x(t_i), y(t_i))$ and $(x(t_{i+1}), y(t_{i+1}))$ on a fine uniform grid with a predefined spatial resolution on $[t_i, t_{i+1}]$ using a cubic curve. If any of interpolated points violates collision avoidance constraints, a local grid refinement procedure is applied to improve the time grid resolution in these segments of trajectories intruding the obstacles. The basic idea of the local grid refinement is to add at least one extra collision-free point on the time interval with collision. In numerical simulations, three collision-free points are added to each piece of trajectory with collision. As illustrated in Fig. 2, since the boundaries

are zig-zag, the first extra point (x_m, y_m) is added to the corner of the obstacle, while the other two points are added at the mid-points between $(x(t_i), y(t_i))$ and (x_m, y_m) , and between (x_m, y_m) and $(x(t_{i+1}), y(t_{i+1}))$, respectively. The time instances associated with the new points are computed based on the length of the path connecting these points and the associated time instances are computed using a constant speed assumption. The new trajectory is obtained by replacing the segment of the old trajectory between $(x(t_i), y(t_i))$ and $(x(t_{i+1}), y(t_{i+1}))$ using the new segment passing through the three new points, as shown as the blue dotted line in Fig. 2.

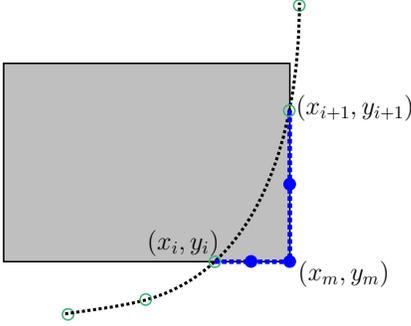


Fig. 2: Local grid refinement.

Numerical simulations show that when the local time grid refinement method is applied after Step 3) of Algorithm 1, most occurrences of inter-collision can be circumvented after 2 or 3 iterations. The local grid refinement however cannot completely avoid inter-collision. To maintain absolute obstacle clearance, one commonly used practice is imposing safety margins on obstacles.

IV. CONVERGENCE AND OPTIMALITY ANALYSIS

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. This assumption is valid, given the mature theory and solvers for QP problems. The following two propositions show that the AQP retains the feasibility of the solution as long as it is started properly, and monotonically improves the energy efficiency of the solution.

Proposition 1: Let \mathcal{X}_i and \mathcal{Y}_i be the motion planning results given by the i^{th} iteration of Algorithm 1. Suppose $(\mathcal{X}_0, \mathcal{Y}_0)$ is feasible for Problem 2, then $(\mathcal{X}_i, \mathcal{Y}_i)$ is feasible for all $i \geq 1$.

Proof: The proposition is shown by induction. Suppose that $(\mathcal{X}_{i-1}, \mathcal{Y}_{i-1})$ is feasible for Problem 2. It follows that constraints (27)-(29) are satisfied. Therefore, we have $X_L \leq D_x(\mathcal{X}_{i-1}; \mathcal{Y}_{i-1}, x_0) \leq X_U$, i.e., \mathcal{X}_{i-1} satisfies the collision avoidance constraints in Problem 3. The other constraints on \mathcal{X} in Problem 3 are the same as (27) of Problem 2, and are automatically satisfied by \mathcal{X}_{i-1} . Therefore, the QP problem in Step 2) of Algorithm 1 has at least one feasible solution \mathcal{X}_{i-1} , hence \mathcal{X}_i , which is the solution given by the QP problem in Step 2), must be feasible.

Similarly, \mathcal{Y}_{i-1} satisfies constraints of Problem 4 and is feasible. Step 3) of Algorithm 1 always yields a feasible solution \mathcal{Y}_i . One verifies that $(\mathcal{X}_i, \mathcal{Y}_i)$ satisfies constraints of Problem 2, thereby establishing the feasibility of $(\mathcal{X}_i, \mathcal{Y}_i)$. ■

Proposition 2: Let \mathcal{X}_i and \mathcal{Y}_i be the 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 $i \rightarrow \infty$.

Proof: Without loss of generality, assume that Algorithm 1 starts from a feasible initial trajectory. The energy cost of the motion planning result after the i^{th} iteration is given by $\tilde{J}(\mathcal{X}_i, \mathcal{Y}_i)$. In Step 2) of the $(i+1)^{\text{th}}$ iteration, the optimal solution \mathcal{X}_{i+1} of the QP Problem 3 minimizes the cost function \tilde{J}_x . Therefore, $\tilde{J}_x(\mathcal{X}_{i+1}) \leq \tilde{J}_x(\mathcal{X})$ for any \mathcal{X} feasible for this problem. It follows that $\tilde{J}_x(\mathcal{X}_{i+1}) \leq \tilde{J}_x(\mathcal{X}_i)$ since \mathcal{X}_i is a feasible solution to this subproblem as shown in the proof of Proposition 1, and \mathcal{X}_{i+1} is the optimal solution. This further implies that $\tilde{J}(\mathcal{X}_{i+1}, \mathcal{Y}_i) = \tilde{J}_x(\mathcal{X}_{i+1}) + \tilde{J}_y(\mathcal{Y}_i) \leq \tilde{J}_x(\mathcal{X}_i) + \tilde{J}_y(\mathcal{Y}_i) = \tilde{J}(\mathcal{X}_i, \mathcal{Y}_i)$. Repeating the same argument for step 3 in the $(i+1)^{\text{th}}$ iteration, we have $\tilde{J}(\mathcal{X}_{i+1}, \mathcal{Y}_{i+1}) \leq \tilde{J}(\mathcal{X}_{i+1}, \mathcal{Y}_i)$. Therefore, $\tilde{J}(\mathcal{X}_{i+1}, \mathcal{Y}_{i+1}) \leq \tilde{J}(\mathcal{X}_i, \mathcal{Y}_i)$, hence the cost function decreases monotonically. Note that by definition, \tilde{J} is bounded by zero. Hence the sequence $\{\tilde{J}(\mathcal{X}_i, \mathcal{Y}_i)\}$ must converge as $i \rightarrow \infty$. ■

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 decoupled dynamics and constraints, Algorithm 1 indeed converges as shown next.

Theorem 4.1: The sequence $(\mathcal{X}_i, \mathcal{Y}_i)$ as given by Algorithm 1 converges as $i \rightarrow \infty$.

Proof: Let $\Delta\mathcal{X}_i = \mathcal{X}_i - \mathcal{X}_{i-1}$, and $\Delta\mathcal{Y}_i = \mathcal{Y}_i - \mathcal{Y}_{i-1}$ for any $i \in \mathbb{N}$. Then it suffices to show that $\|\Delta\mathcal{X}_i\| \rightarrow 0$ and $\|\Delta\mathcal{Y}_i\| \rightarrow 0$ as $i \rightarrow \infty$. Suppose this is not true, then there exists $\delta > 0$ such that for any $j \in \mathbb{N}$, there exist some $k > j$, $k \in \mathbb{N}$ with either $\|\Delta\mathcal{X}_k\| > \delta$ or $\|\Delta\mathcal{Y}_k\| > \delta$, or both. Since $\tilde{J}(\mathcal{X}_i, \mathcal{Y}_i)$ decreases monotonically according to Proposition 2, we may choose j large enough such that $\tilde{J}(\mathcal{X}_{k-1}, \mathcal{Y}_{k-1}) - \tilde{J}(\mathcal{X}_k, \mathcal{Y}_k) < \epsilon$ for an arbitrarily small positive number ϵ .

Without loss of generality, assume that $\|\Delta\mathcal{X}_k\| > \delta$. Note that $\tilde{J}(\mathcal{X}_k, \mathcal{Y}_{k-1}) = \tilde{J}(\mathcal{X}_{k-1} + \Delta\mathcal{X}_k, \mathcal{Y}_{k-1}) = \tilde{J}_x(\mathcal{X}_{k-1} + \Delta\mathcal{X}_k) + \tilde{J}_y(\mathcal{Y}_{k-1})$, and $\tilde{J}(\mathcal{X}_{k-1}, \mathcal{Y}_{k-1}) = \tilde{J}_x(\mathcal{X}_{k-1}) + \tilde{J}_y(\mathcal{Y}_{k-1})$ by (32). Since $\tilde{J}(\mathcal{X}_k, \mathcal{Y}_k) \leq \tilde{J}(\mathcal{X}_k, \mathcal{Y}_{k-1}) \leq \tilde{J}(\mathcal{X}_{k-1}, \mathcal{Y}_{k-1})$ following Proposition 2, we have

$$0 \leq \tilde{J}_x(\mathcal{X}_{k-1}) - \tilde{J}_x(\mathcal{X}_{k-1} + \Delta\mathcal{X}_k) < \epsilon.$$

Since \tilde{J}_x is a quadratic function, the following relation holds by the Taylor series expansion at \mathcal{X}_{k-1} :

$$\begin{aligned} \tilde{J}_x(\mathcal{X}_{k-1} + \Delta\mathcal{X}_k) &= \tilde{J}_x(\mathcal{X}_{k-1}) + \nabla\tilde{J}_x(\mathcal{X}_{k-1})\Delta\mathcal{X}_k \\ &\quad + \frac{1}{2}[\Delta\mathcal{X}_k]^T \nabla^2\tilde{J}_x(\mathcal{X}_{k-1})[\Delta\mathcal{X}_k], \end{aligned}$$

from which we have

$$\begin{aligned} -\epsilon &< \tilde{J}_x(\mathcal{X}_{k-1} + \Delta\mathcal{X}_k) - \tilde{J}_x(\mathcal{X}_{k-1}) \\ &= \nabla\tilde{J}_x(\mathcal{X}_{k-1})\Delta\mathcal{X}_k + \frac{1}{2}[\Delta\mathcal{X}_k]^T \nabla^2\tilde{J}_x(\mathcal{X}_{k-1})[\Delta\mathcal{X}_k] \leq 0, \end{aligned} \quad (35)$$

where $\nabla\tilde{J}_x(\mathcal{X}_{k-1})$ and $\nabla^2\tilde{J}_x(\mathcal{X}_{k-1})$ are the Jacobian and Hessian matrices, respectively. Because all constraints of the QP problem solved in the second step of Algorithm 1 are convex, and both \mathcal{X}_{k-1} and $\mathcal{X}_{k-1} + \Delta\mathcal{X}_k$ are feasible solutions according to Proposition 1, it follows that $\mathcal{X}_{k-1} + \frac{1}{2}\Delta\mathcal{X}_k$ is also

a feasible solution to this QP problem. Because $\mathcal{X}_{k-1} + \Delta\mathcal{X}_k$ is the optimal solution, we have $\tilde{J}_x(\mathcal{X}_{k-1} + \frac{1}{2}\Delta\mathcal{X}_k) \geq \tilde{J}_x(\mathcal{X}_{k-1} + \Delta\mathcal{X}_k) > \tilde{J}_x(\mathcal{X}_{k-1}) - \epsilon$. Meanwhile, the following is true due to the convexity of \tilde{J}_x and the optimality of $\mathcal{X}_{k-1} + \Delta\mathcal{X}_k$:

$$\begin{aligned} \tilde{J}_x(\mathcal{X}_{k-1} + \frac{1}{2}\Delta\mathcal{X}_k) &\leq \frac{1}{2}\tilde{J}_x(\mathcal{X}_{k-1} + \Delta\mathcal{X}_k) + \frac{1}{2}\tilde{J}_x(\mathcal{X}_{k-1}) \\ &\leq \tilde{J}_x(\mathcal{X}_{k-1}). \end{aligned}$$

Therefore the following is true

$$-\epsilon < \tilde{J}_x(\mathcal{X}_{k-1} + \frac{1}{2}\Delta\mathcal{X}_k) - \tilde{J}_x(\mathcal{X}_{k-1}) \leq 0.$$

Taking the Taylor series expansion of $\tilde{J}_x(\mathcal{X}_{k-1} + \frac{1}{2}\Delta\mathcal{X}_k)$ at \mathcal{X}_{k-1} , the above expression can be written as

$$-8\epsilon < 4\nabla\tilde{J}_x(\mathcal{X}_{k-1})\Delta\mathcal{X}_k + [\Delta\mathcal{X}_k]^T \nabla^2\tilde{J}_x(\mathcal{X}_{k-1})[\Delta\mathcal{X}_k] \leq 0. \quad (36)$$

Because ϵ is arbitrarily small, (35)-(36) hold at the same time only if $[\Delta\mathcal{X}_k]^T \nabla^2\tilde{J}_x(\mathcal{X}_{k-1})[\Delta\mathcal{X}_k] = 0$. However, since $\nabla^2\tilde{J}_x(\mathcal{X}_{k-1})$ is strictly positive definite, we have $\Delta\mathcal{X}_k = 0$, which is a contradiction to $\|\Delta\mathcal{X}_k\| > \delta$. Proof is complete. ■

Theorem 4.2: Suppose that Algorithm 1 converges to a feasible solution $(\mathcal{X}^*, \mathcal{Y}^*)$. Assume that the boundary of the collision-free region \mathcal{D} is composed of piecewise linear segments which are parallel to either the x or y axis. Then $(\mathcal{X}^*, \mathcal{Y}^*)$ is a locally optimal solution to Problem 2.

Proof: First, consider the case when the nodes corresponding to $(\mathcal{X}^*, \mathcal{Y}^*)$ do not coincide with any corner (non-differentiable) points of the feasible region. In such a case, the collision-avoidance constraints for $(\mathcal{X}^*, \mathcal{Y}^*)$ are decoupled in its neighborhood, since at any node lying on the boundary of \mathcal{D} , the collision avoidance constraint is either a bound on the x -coordinate of the node only, or on the y -coordinate of the node only. Furthermore, the collision avoidance constraint is differentiable.

For convenience, denote the constraints in Problem 2 as

$$\begin{aligned} c_n(\mathcal{X}, \mathcal{Y}) &= 0, \quad n \in \mathcal{E} \\ c_n(\mathcal{X}, \mathcal{Y}) &\leq 0, \quad n \in \mathcal{I}, \end{aligned}$$

where \mathcal{E} and \mathcal{I} are the index sets of equality, and inequality constraints in Problem 2, respectively. Similarly, let \mathcal{E}_x and \mathcal{I}_x denote the index sets of equality and inequality constraints in Problem 3, and let \mathcal{E}_y and \mathcal{I}_y denote the index sets of equality and inequality constraints in Problem 4. Because the constraints are decoupled for the x and y motions, we have $\mathcal{E} = \mathcal{E}_x \cup \mathcal{E}_y$, $\mathcal{E}_x \cap \mathcal{E}_y = \emptyset$, $\mathcal{I} = \mathcal{I}_x \cup \mathcal{I}_y$, and $\mathcal{I}_x \cap \mathcal{I}_y = \emptyset$, i.e., if a constraint is active in Problem 2, then it is also active in either Problem 3, or Problem 4, but not both.

Since $(\mathcal{X}^*, \mathcal{Y}^*)$ is a convergent solution of Algorithm 1, it is also the solution in the second and third step of Algorithm 1. Specifically, with the fixed y -motion given by \mathcal{Y}^* , \mathcal{X}^* is the optimal solution to Problem 3. Hence, the Karush-Kuhn-Tucker (KKT) conditions for Problem 3 are satisfied by \mathcal{X}^* :

$$\nabla_x \mathcal{L}_x(\mathcal{X}^*, \lambda_x^*) = 0 \quad (37)$$

$$c_n(\mathcal{X}^*, \mathcal{Y}^*) = 0, \quad n \in \mathcal{E}_x \quad (38)$$

$$c_n(\mathcal{X}^*, \mathcal{Y}^*) \leq 0, \quad n \in \mathcal{I}_x \quad (39)$$

$$\lambda_{x_n}^* \leq 0, \quad n \in \mathcal{I}_x, \quad (40)$$

$$\lambda_{x_n}^* c_n(\mathcal{X}^*, \mathcal{Y}^*) = 0, \quad n \in \mathcal{I}_x \cup \mathcal{E}_x, \quad (41)$$

where

$$\mathcal{L}_x(\mathcal{X}, \lambda_x) = \tilde{J}_x(\mathcal{X}) + \sum_{n \in \mathcal{E}_x \cup \mathcal{I}_x} \lambda_{x_n} c_n(\mathcal{X}, \mathcal{Y}^*). \quad (42)$$

The KKT conditions for Problem 4 are satisfied by \mathcal{Y}^* :

$$\nabla_y \mathcal{L}_y(\mathcal{Y}^*, \lambda_y^*) = 0 \quad (43)$$

$$c_n(\mathcal{X}^*, \mathcal{Y}^*) = 0, \quad n \in \mathcal{E}_y \quad (44)$$

$$c_n(\mathcal{X}^*, \mathcal{Y}^*) \leq 0, \quad n \in \mathcal{I}_y \quad (45)$$

$$\lambda_{y_n}^* \leq 0, \quad n \in \mathcal{I}_y \quad (46)$$

$$\lambda_{y_n}^* c_n(\mathcal{X}^*, \mathcal{Y}^*) = 0, \quad n \in \mathcal{I}_y \cup \mathcal{E}_y, \quad (47)$$

where

$$\mathcal{L}_y(\mathcal{Y}, \lambda_y) = \tilde{J}_y(\mathcal{Y}) + \sum_{n \in \mathcal{E}_y \cup \mathcal{I}_y} \lambda_{y_n} c_n(\mathcal{X}^*, \mathcal{Y}). \quad (48)$$

The Lagrangian for Problem 2 is given by

$$\mathcal{L}(\mathcal{X}, \mathcal{Y}, \lambda) = \tilde{J}(\mathcal{X}, \mathcal{Y}) + \sum_{n \in \mathcal{E} \cup \mathcal{I}} \lambda_n c_n(\mathcal{X}, \mathcal{Y}^*). \quad (49)$$

Let

$$\lambda_n^* = \begin{cases} \lambda_{x_n}^*, & n \in (\mathcal{E}_x \cup \mathcal{I}_x) \\ \lambda_{y_n}^*, & n \in (\mathcal{E}_y \cup \mathcal{I}_y), \end{cases}$$

According to the definition of the cost function (32), we have

$$\nabla_x \tilde{J}(\mathcal{X}^*, \mathcal{Y}^*) = \nabla_x \tilde{J}_x(\mathcal{X}^*)$$

$$\nabla_y \tilde{J}(\mathcal{X}^*, \mathcal{Y}^*) = \nabla_y \tilde{J}_y(\mathcal{Y}^*).$$

Besides, since the collision-avoidance constraints are composed of linear segments parallel to either the x or y axis, the collision avoidance constraint on $(\mathcal{X}, \mathcal{Y})$ at $(\mathcal{X}^*, \mathcal{Y}^*)$ is equivalent to either bounds on \mathcal{X} or bounds on \mathcal{Y} . Then it can be verified that

$$\nabla \mathcal{L}(\mathcal{X}^*, \mathcal{Y}^*, \lambda^*) = [\nabla_x \mathcal{L}_x(\mathcal{X}^*, \lambda_x^*), \nabla_y \mathcal{L}_y(\mathcal{Y}^*, \lambda_y^*)] = 0.$$

Furthermore, it can also be verified that the other KKT conditions at $(\mathcal{X}^*, \mathcal{Y}^*)$ for Problem 2 also hold according to (38) to (41), and (44) to (47). The procedure is straightforward, and the details are omitted for brevity. Hence, the first order optimality condition is satisfied at $(\mathcal{X}^*, \mathcal{Y}^*)$. Because \tilde{J} is strictly convex, the second order condition also holds. Therefore $(\mathcal{X}^*, \mathcal{Y}^*)$ is at least a local optimal solution.

Now consider the case when one of the nodes, say, the i^{th} node in the trajectory corresponding to $(\mathcal{X}^*, \mathcal{Y}^*)$ coincides with a corner point P of the feasible region \mathcal{D} .

Because the collision-avoidance constraint is not differentiable at P , the previous proof does not apply directly. We can decompose the feasible region around P as a union of two subregions, as illustrated in Fig. 3. When restricting the node in each of the subregions, it follows from the first part of the proof that $(\mathcal{X}^*, \mathcal{Y}^*)$ is optimal. Hence, $(\mathcal{X}^*, \mathcal{Y}^*)$ is optimal when the node lies in the union of the two subregions, i.e., $(\mathcal{X}^*, \mathcal{Y}^*)$ is optimal for the original problem. When there are multiple points on the corner of \mathcal{D} , then by decomposing the feasible neighborhood of $(\mathcal{X}^*, \mathcal{Y}^*)$ in a similar way, and consider all combinations of the feasible region, the optimality of $(\mathcal{X}^*, \mathcal{Y}^*)$ can be proved. ■

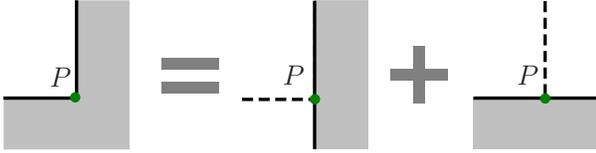


Fig. 3: Feasible region decomposition.

According to Theorem 4.2, the proposed AQP algorithm outputs an optimal solution when the boundary of the collision-free region is a concatenation of zigzag lines parallel to the x or y axis. As shown in Fig. 4 below, obstacles with complex boundaries in the 2D plane can be approximated by union of rectangles, yielding the zig-zag type of boundary required by Theorem 4.2. The approximation error can be driven small by increasing the number of rectangles as well as the grid size N , at the expense of computational efficiency. Obstacles in the 3D space can be approximated similarly using rectangular boxes.

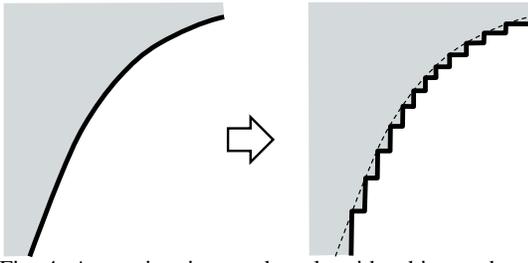


Fig. 4: Approximation to obstacle with arbitrary shape

V. EXAMPLES

This section performs simulation study to validate the energy efficiency of the solution result from AQP, by comparing with results from solving an MIQP problem. Computational efficiency and effectiveness of the proposed algorithm is further verified by applying to more complex examples. In simulation, the parameters in the system dynamics and cost function are given by: $d_{x,y} = 14.03$, $b_{x,y} = 3781.9$, $R_{x,y} = 5.06$, $Q_{x,y} = 1.01E - 3$, $v_{x,y_{\max}} = -v_{x,y_{\min}} = 314.16\text{rad/s}$, and $u_{x,y_{\max}} = -u_{x,y_{\min}} = 3$.

A. Optimality

The non-convex region \mathcal{D} can be approximated and decomposed into a finite number of polygons described by linear constraints. By associating each polygon with a binary variable indicating whether the trajectory is inside this polygon, the collision avoidance constraints in Problem 2 can be formulated as mixed-integer linear constraints. Therefore Problem 2 can be solved by MIQP solvers. When its solutions converge, the MIQP solver produces the global optimal solution.

The test case shown in Figure 5 presents a cornering problem typical for the motion planning of CNC machines and robots. The gray region is the obstacle. The final time t_f , initial position (x_0, y_0) , final position (x_f, y_f) , clearance ℓ_x and ℓ_y are specified differently in a total of 9 test cases. Both initial and final speeds are zero. In both algorithms, Problem

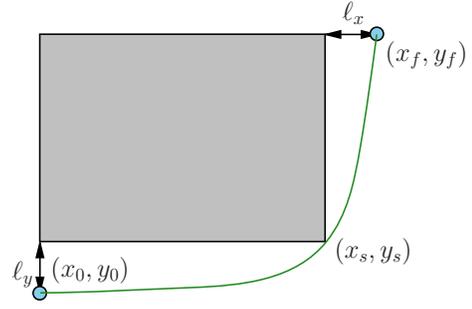


Fig. 5: Test case geometry setup for AQP and MIQP trajectory planning comparison

TABLE I: Performance comparison of the AQP and MIQP

Case #	Energy saving*		CPU time (s)	
	AQP	MIQP	AQP	MIQP
1	35.2%	40.3%	0.1202	57.65
2	21.9%	28.4%	0.1067	212.89
3	55.3%	58.5%	0.0961	98.08
4	61.3%	63.0%	0.1006	150.73
5	77.4%	78.3%	0.0873	59.67
6	53.5%	57.0%	0.0847	163.60
7	32.2%	38.7%	0.1047	166.66
8	33.6%	41.5%	0.1050	62.28
9	35.0%	41.7%	0.1031	75.96
Average	45.0%	49.7%	0.1009	116.4

*relative to heuristic trajectory with trapezoidal speed profiles

1 is discretized on a uniform time grid of 60 points. The local grid refinement algorithm is not used such that AQP maintains the same time grid resolution as MIQP for fair comparison.

The initial guess trajectory for starting the AQP is generated based on a trapezoidal speed profile. Specifically, the initial guess comprises two segments of position transition trajectories, with the first segment between (x_0, y_0) and (x_s, y_s) , and the second segment between (x_s, y_s) and (x_f, y_f) . The net travel distances of two segments are $L_1 = \sqrt{(x_s - x_0)^2 + (y_s - y_0)^2}$ and $L_2 = \sqrt{(x_s - x_f)^2 + (y_s - y_f)^2}$, and the corresponding travel time is specified as $t_f L_1 / (L_1 + L_2)$ and $t_f L_2 / (L_1 + L_2)$, respectively. Subsequently, a collision free trajectory satisfying all constraints is obtained by assigning trapezoidal speed profiles for x motion and y motion independently to produce desired position transition from (x_0, y_0) to (x_f, y_f) . Simulation results are shown in Table I. The AQP-based motion planning can achieve lower but comparable energy efficiency as MIQP, albeit at a significantly lower computation load.

B. Computation Efficiency

The AQP is applied to solve more complex collision-free trajectory planning problems, which are computationally prohibitive for the MIQP approach. The system motion is contained by tunnels depicted as white space in Fig. 6 and Fig. 8, in which the gray region represents obstacles. The initial guesses of AQP are generated by first calculating the minimum time trajectories connecting the corner points in these figures with zero initial and final speeds, then relaxing the travel time by a certain factor greater than 1 to help with

energy-saving. The energy-efficient paths, shown in Fig. 6 and Fig. 8, are generated by AQP using two different scenarios, where the solid blue curves correspond to problem data used in [27], and the dash red curves correspond to problem data in this paper. Due to more restrictive control inputs in this work, the dash red are smoother than the solid blue, for both cases. The energy-efficient trajectories satisfy collision avoidance constraints. From the resultant speed and control trajectories, one can readily verify that the speed and control constraints are satisfied for both cases. Figs. 7 and 9 plot the speed profiles which verify the velocity constraint. Combining the speed profiles and path figures, one can observe that an energy-efficient trajectory leads to energy saving by avoiding stops, though increasing the length of the corresponding path.

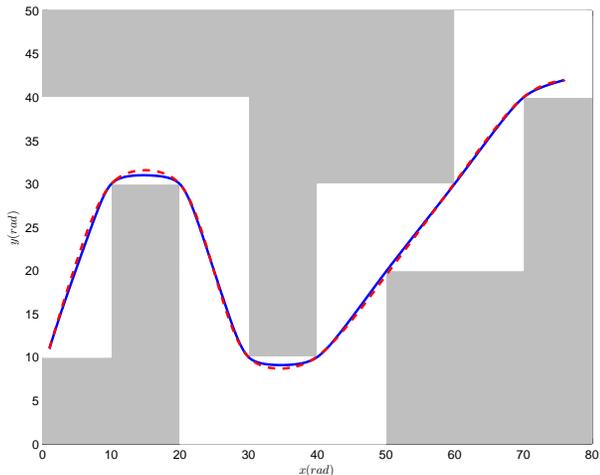


Fig. 6: Energy-efficient paths, case 1.

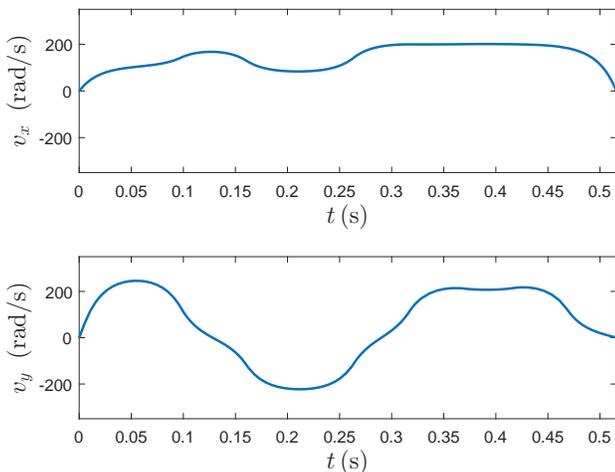


Fig. 7: Speed profile, case 1.

In order to better ensure collision avoidance between grid points, the grid refinement scheme in Section III-E is applied to adaptively refine the time grid when collision occurs between neighboring grid points. The behavior of the local grid refinement technique is shown in Figure 10, which depicts trajectories near position [40, 40] from 3 consecutive iterations when applying the AQP and grid refinement algorithms for case 2.

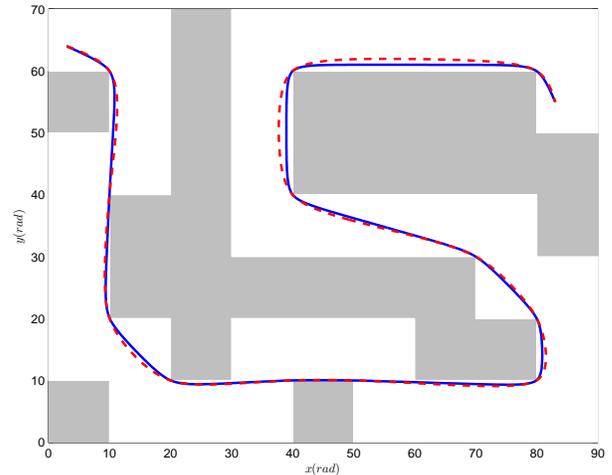


Fig. 8: Energy-efficient paths, case 2.

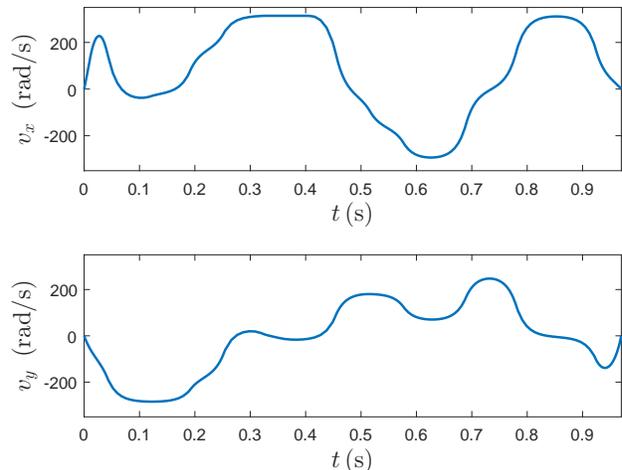


Fig. 9: Speed profile, case 2.

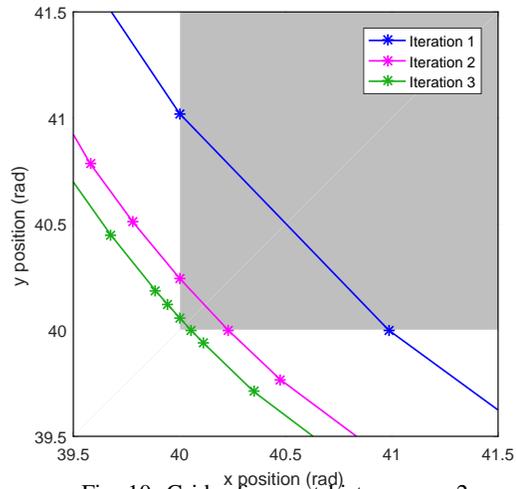


Fig. 10: Grid refinement history, case 2.

TABLE II: Energy consumption comparison.

		J_x	J_y	J	Energy saving	$T_{CPU}(s)$
Case 1	Heuristic	86.5	59.3	145.8	0	–
	AQP	51.1	26.4	77.5	46.8%	0.69
Case 2	Heuristic	32.9	30.8	63.7	0	–
	AQP	23.8	18.0	31.8	50.1%	2.91

Table II compares the energy consumption of trajectories shown in Figs. 6 and 8. The AQP-based motion planning is capable of substantially reducing the energy consumption in both x and y directions. Given an optimization problem and a numerical solver, the computation time depends on the property of the cost function, e.g. conditionedness of the Jacobian and Hessian, and the problem scale, which is precisely captured by the grid size N . For case 1, $N = 148$ when the algorithm terminates. For case 2, the final grid size is $N = 254$. For both cases AQP stops after three iterations including two local grid refinements.

VI. CONCLUSION

This paper studied energy-efficient trajectory generation for multi-axis motion systems with decoupled linear time-invariant (LTI) dynamics. Resorting to numerical optimization, the problem is reduced to non-convex programming problem, due to the existence of obstacles and the non-convex cost function. An Alternating Quadratic Programming (AQP) method was proposed to solve the problem with convergence guarantee. AQP is enabled by the following two key observations: system dynamics can play a conducive role to convexify a non-convex cost function; non-convex obstacles can be abstracted as linear constraints through the decomposition technique. In the case that the boundaries of obstacles are parallel to the motion axes, the AQP leads to a local optimal solution. Compared with the treatment using integer variables to cope with obstacles, the AQP is, although sub-optimal, computationally efficient and suitable for more complex motion planning applications.

Starting with an initial feasible trajectory, AQP merely searches within the same homotopy class of trajectories for a solution with improved energy efficiency. In order to achieve global optimality, AQP needs to work with another module which provides exhaustive classes of feasible trajectories. Aiming to secure convergence guarantee and local optimality, this work imposes restrictive assumptions on system dynamics and obstacles, which again indicates future work. For instance, is it possible to relax the requirement that the system dynamics have to be LTI and decoupled; how to generalize this work to deal with dynamic obstacles and cooperative trajectory planning involving multiple agents.

VII. ACKNOWLEDGEMENT

This work was done while the first author was affiliated with Mitsubishi Electric Research Laboratories. Authors would like to thank editor and reviewers for valuable comments to improve this work.

REFERENCES

- [1] P. Tokekar, N. Karnad, and V. Isler, "Energy-optimal trajectory planning for car-like robots," *Autonomous Robots*, vol. 37, no. 3, pp. 279–300, 2014. [Online]. Available: <http://dx.doi.org/10.1007/s10514-014-9390-3>
- [2] B. R. Donald and P. Xavier, "Provably good approximation algorithms for optimal kinodynamic planning: robots with decoupled dynamics bounds," *Algorithmica*, vol. 14, pp. 443–479, 1995.
- [3] S. Liu and D. Sun, "Minimizing energy consumption of wheeled mobile robots via optimal motion planning," *IEEE/ASME Trans. Mechatron.*, vol. 19, no. 2, pp. 401–411, April 2014.
- [4] Z. Sun and J. H. Reif, "On finding energy-minimizing paths on terrains," *IEEE Trans. Robot.*, vol. 21, no. 1, pp. 102–114, Feb 2005.
- [5] A. Stentz, "Optimal and efficient path planning for unknown and dynamic environments," Robotics Institute, Carnegie Mellon University, Tech. Rep. CMU-RI-TR-93-20, 1993.
- [6] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning A*," *Artificial Intelligence*, vol. 155, no. 1-2, pp. 93–146, 2004.
- [7] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.
- [8] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robot Res.*, vol. 20, no. 5, pp. 378–400, 2001.
- [9] S. M. LaValle, *Planning Algorithms*. Cambridge, UK: Cambridge University Press, 2006.
- [10] Y. Wang, K. Ueda, and S. A. Bortoff, "A Hamiltonian approach to compute an energy efficient trajectory for a servomotor system," *Automatica*, vol. 49, no. 12, pp. 3550–3561, Dec. 2013.
- [11] Y. Wang, Y. Zhao, S. A. Bortoff, and K. Ueda, "A real-time energy-optimal trajectory generation method for a servomotor system," *IEEE Trans. Ind. Electron.*, vol. 62, no. 2, pp. 1175–1188, Feb. 2015.
- [12] O. Wigstrom, B. Lennartson, A. Vergnano, and C. Breitholtz, "High-level scheduling of energy optimal trajectories," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 1, pp. 57–64, 2013.
- [13] Y. Zhao and P. Tsiotras, "Analysis of energy-optimal aircraft landing operation trajectories," *J. Guid. Control Dynam.*, vol. 36, no. 3, pp. 833–845, 2013.
- [14] T. Chettibi, H. E. Lehtihet, M. Haddad, and S. Hanchi, "Minimum cost trajectory planning for industrial robots," *European Journal of Mechanics A: Solids*, vol. 23, no. 4, pp. 703–715, 2004.
- [15] S. Bjorkenstam, D. Gleeson, R. Bohlin, C. S. Johan, and B. Lennartson, "Energy efficient and collision free motion of industrial robots using optimal control," in *Proc. 2013 CASE*, 2013, pp. 510–515.
- [16] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *Int. J. Robot Res.*, vol. 28, no. 8, pp. 933–945, 2009.
- [17] D. Hsu, R. Kindel, J. C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *Int. J. Robot Res.*, vol. 21, no. 3, pp. 233–255, 2002.
- [18] J. T. Betts, "Survey of numerical methods for trajectory optimization," *J. Guid. Control Dynam.*, vol. 21, no. 2, pp. 193–207, Mar.-Apr. 1998.
- [19] G. Elmagar, M. A. Kazemi, and M. Razzaghi, "The pseudospectral legendre method for discretizing optimal control problems," *IEEE Trans. Automat. Control*, vol. 40, no. 10, pp. 1793–1796, Oct. 1995.
- [20] H. Yu, Y. Wang, S. A. Bortoff, and K. Ueda, "Energy-efficient trajectory planning for a mobile agent by using a two-stage decomposition approach," in *IFAC Proceedings*, vol. 47, 2014, pp. 3851–3856.
- [21] A. Richards, T. Schouwenaars, J. P. How, and E. Feron, "Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming," *J. Guid. Control Dynam.*, vol. 25, no. 4, pp. 755–763, July-August 2002.
- [22] D. Mellinger, A. Kushleyev, and V. Kumar, "Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams," in *Proc. 2012 ICRA*, St. Paul, MN, May 14-18 2012, pp. 477 – 483.
- [23] K. G. Shin and N. D. McKay, "A dynamic programming approach to trajectory planning of robotic manipulators," *IEEE Trans. Automat. Control*, vol. AC-31, no. 6, pp. 491–500, Jun. 1986.
- [24] S. Singh and M. C. Leu, "Optimal trajectory generation for robotic manipulators using dynamics programming," *Trans. ASME, J. Dyn. Sys. Meas. Control*, vol. 109, pp. 88–96, Jun. 1987.
- [25] D. Verschuere, B. Demeulenaere, J. Swevers, J. D. Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Trans. Automat. Control*, vol. 54, no. 10, pp. 2318–2327, Oct. 2009.
- [26] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot Res.*, vol. 30, no. 7, pp. 846–894, 2011.
- [27] Y. Zhao, Y. Wang, S. A. Bortoff, and D. Nikovski, "Energy-efficient collision-free trajectory planning using alternating quadratic programming," in *Proc. 2014 ACC*, Portland, OR, Jun. 2014, pp. 1249–1254.
- [28] A. Staglianò, G. Chiusano, C. Basso, and M. Santoro, *Learning Adaptive and Sparse Representations of Medical Images*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 130–140.
- [29] L. Xiao, M. Johansson, H. Hindi, S. Boyd, and A. Goldsmith, *Joint Optimization of Wireless Communication and Networked Control Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 248–272.

- [30] N. Uchiyama, Y. Ogawa, A. E. K. M., and S. Sano, "Energy saving in five-axis machine tools using synchronous and contouring control and verification by machining experiment," *IEEE Trans. Ind. Electron.*, vol. 62, no. 9, pp. 5608–5618, Sept 2015.
- [31] M. Duan and C. E. Okwudire, "Energy-efficient controller design for a redundantly actuated hybrid feed drive with application to machining," *IEEE/ASME Trans. Mechatron.*, vol. 21, no. 4, pp. 1822–1834, Aug 2016.
- [32] A. Sciarretta, G. D. Nunzio, and L. L. Ojeda, "Optimal ecodriving control: Energy-efficient driving of road vehicles as an optimal control problem," *IEEE Control Systems*, vol. 35, no. 5, pp. 71–90, Oct 2015.
- [33] 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.
- [34] C. Shih, T. Lee, and W. Gruver, "Motion planning with time-varying polyhedral obstacles based on graph search and mathematical programming," in *Proc. 1990 ICRA*, vol. 1, Cincinnati, OH, USA, May 1990, pp. 331 – 337.
- [35] O. Brock and L. E. Kavraki, "Decomposition-based motion planning: a framework for real-time motion planning in high-dimensional configuration spaces," in *Proc. 2001 ICRA*, vol. 2, May 2001, pp. 1469 – 1474.
- [36] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Trans. Automat. Control*, vol. AC-30, no. 6, pp. 531–541, Jun. 1985.
- [37] 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.
- [38] 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.

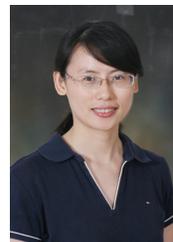


MengChu Zhou (S'88-M'90-SM'93-F'03) received his B.S. degree in Control Engineering from Nanjing University of Science and Technology, Nanjing, China in 1983, M.S. degree in Automatic Control from Beijing Institute of Technology, Beijing, China in 1986, and Ph. D. degree in Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY in 1990. He joined New Jersey Institute of Technology (NJIT), Newark, NJ in 1990, and is now a Distinguished Professor of Electrical and Computer Engineering. His research interests are in Petri nets, intelligent automation, Internet of Things, big data, web services, and intelligent transportation. He has over 700 publications including 12 books, 400+ journal papers (300+ in IEEE transactions), 11 patents and 28 book-chapters. He is the founding Editor of IEEE Press Book Series on Systems Science and Engineering and Editor-in-Chief of IEEE/CAA Journal of Automatica Sinica. He is a recipient of Humboldt Research Award for US Senior Scientists from Alexander von Humboldt Foundation, Franklin V. Taylor Memorial Award and the Norbert Wiener Award from IEEE Systems, Man and Cybernetics Society for which he serves as VP for Conferences and Meetings. He is a life member of Chinese Association for Science and Technology-USA and served as its President in 1999. He is a Fellow of International Federation of Automatic Control (IFAC), American Association for the Advancement of Science (AAAS) and Chinese Association of Automation (CAA).



control of directional drilling tools.

Yiming Zhao (M'12) received his Ph.D. in Aerospace Engineering (2012) and M.S. in Mathematics (2011) from Georgia Institute of Technology. He also received B.S. and M.S. degrees in Aerospace Engineering from Beijing University of Aeronautics and Astronautics, China. He joined Haliburton energy services since 2013. Previously he was with Mitsubishi Electric Research Laboratories, Cambridge, MA. His recent research finds applications in active vibration control of acoustic logging tool, drilling vibration mitigation and autonomous



control systems.

Jing Wu (M'08) received her B.S. degree from Nanchang University in 2000, M.S. degree from Yanshan University in 2002, and Ph.D. degree from University of Alberta in 2008, all in electrical engineering. Since 2011, she has been with Shanghai Jiao Tong University, Shanghai, China, and is currently an Associate Professor. Dr. Wu is a registered Professional Engineer in Alberta, Canada. Her current research interests include robust model predictive control, security control, stability analysis and estimation for cyber-physical systems, smart grids and networked



Engineer, Project Manager, and Manager of R&D Dept. in industries. His research interests include nonlinear control and estimation, optimal control, adaptive systems and their applications including mechatronic systems.

Yebin Wang (S'06-M'10-SM'16) received his B.Eng. degree in Mechatronics Engineering from Zhejiang University, Hangzhou, China in 1997, M.Eng. degree in Control Theory & Control Engineering from Tsinghua University, Beijing, China in 2001, and Ph.D. in Electrical Engineering from the University of Alberta, Edmonton, Canada in 2008. Dr. Wang has been with Mitsubishi Electric Research Laboratories in Cambridge, MA, USA since 2009, and now is a Senior Principal Research Scientist. From 2001 to 2003 he was an Automation