A Unified Formulation and Nonconvex Optimization Method for Mixed-Type Decision-Making of Robotic Systems

Chuangchuang Sun, Nathaniel Kingry, and Ran Dai

Abstract—**Mixed-type decision-making is ubiquitously required in robotic systems and has attracted significant research interests. Examples include, but not limited to, the integrated task and motion planning and optimal control of hybrid systems involving both continuous and discrete dynamic behaviors. For decision-making of robotic systems to improve operational efficiency, safety, and/or mission success rate, they involve both discrete variables representing task allocation or transitions between discrete modes and continuous variables representing trajectories of the planned motion or states governed by differential equations. This paper formulates a class of mixed-type decision-making problems with polynomial objective and constraints as quadratically constrained quadratic programming (QCQP) problems and a nonconvex optimization method based on alternating direction method of multipliers is proposed to solve the QCQP. The proposed optimization method consists of three sequential subproblems, all of which admit closed-form solutions. Moreover, convergence proof of the optimization algorithm is provided. Two representative problems, traveling salesman with obstacle avoidance and rendezvous and docking of a charging station with distinct phase constraints, are described and solved via the proposed method. Numerical simulations as well as experimental verification of both problems are presented and compared with a state-of-art method to validate the effectiveness, efficacy and robustness of the nonconvex optimization method.**

Index Terms—**Alternating direction method of multipliers (ADMMs), hybrid control, integrated task and motion planning (ITMP), mixed-type decision-making, quadratically constrained quadratic programming, robot rendezvous and docking.**

I. Introduction

Mixed-type decision-making, consisting of both discrete and continuous decision variables, are often found in a variety of engineering problems, especially for motion planning and control of robotic systems. When employing optimal decision-making strategies, many robotic systems benefit from performance and resilience improvement that significantly increases autonomy level, maneuverability, multitask capability. However, due to the enormous decision space and complex nature of robotic systems, determining the optimal solution for both discrete and continuous variables suffers from the curse of dimensionality. Two types of representative mixed-type decision-making problems and their applications in robotic systems are introduced here, including integrated task and motion planning (ITMP) and a class of optimal control of hybrid systems.

A representative ITMP problem considered here requires an unmanned ground vehicle (UGV) to visit a group of waypoints exactly once (and not necessarily to return to the starting point) in the shortest distance or time. While it resembles the classic traveling salesman problem (TSP), the problem considered here is significantly distinct from the traditional TSP in twofolds. At first, as obstacles are considered, the distance between a pair of waypoints is no longer the Euclidean distance. When simply skipping the straight paths going through the obstacles, it will sacrifice optimality and on some occasions can even lead to infeasible solutions. More importantly, when considering the kinematic constraints of a UGV, high turning rates are usually not allowed, which indicates the vehicle is unable to turn sharply and quickly point itself directly to the next waypoint. Intuitively, it requires that the planned motion be relatively smooth, especially at the transition of the waypoints and around the obstacles. Consequently, we aim to solve a kinematically feasible and obstacle-free TSP.

The other representative decision-making problem is the optimal control of robot rendezvous and docking problem. The focus of this optimal hybrid control problem is to search the optimal path of a mobile robot to approach a charging station with desired states at the terminal point. That is to accomplish the rendezvous and docking mission within minimum time or using minimum control efforts while satisfying the operational constraints along the path and achieving high precision docking at the charging station. As a result, we divide the rendezvous and docking operation into two phases. In the first phase, the robot can move at a relatively faster speed and low precision to the neighborhood of the charging station. In the second phase, the robot is constrained in a proximity zone at a lower speed. The purpose of employing two-phase operational modes are to balance the optimality and precision, which naturally leads to

Manuscript received October 4, 2020; accepted November 2, 2020. Date of publication November 23, 2020; date of current version June 4, 2021. This work was supported by NSF Grants ECCS-1815930. This paper was recommended for publication by Associate Editor R. Vasudevan and Editor P. Robuffo Giordano upon evaluation of the reviewers' comments. *(Corresponding author: Ran Dai.)*

Chuangchuang Sun is with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: ccsun1@mit.edu).

Nathaniel Kingry is with the Rockwell Collins, Cedar Rapids, IA 52498 USA (e-mail: nkingry@iastate.edu).

Ran Dai is with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47907 USA (e-mail: randai@purdue.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TRO.2020.3036619.

Digital Object Identifier 10.1109/TRO.2020.3036619

an optimal hybrid control problem. We need to find the optimal path along both phases, as well as, the optimal transition point from phase one to phase two.

We formulate a class of mixed-type decision-making problems with polynomial objective and dynamics in a compact form integrating task-level decisions or decisions on dynamical mode and continuous decisions on states into one whole problem. Moreover, the formulation incorporates the dynamical constraints as well as the mission/operational constraints via discretization and transformation of trigonometric functions. After introducing binary variables and handling them as continuous ones with quadratic constraints, a general mixed-type decision-making problem is reformulated as a quadratically constrained quadratic program (QCQP) problem. Furthermore, a novel optimization algorithm based on alternating direction method of multipliers (ADMM) is proposed to solve the QCQPs.

A. *Related Work*

Comprehensive robot maneuvers require not only navigating robots toward feasible paths, but also achieving a set of objects in an efficient order and optimized performance index. Such ITMP mission is composed of discrete decisions for task allocation and continuous decisions for planned paths. Efficient algorithms have been developed to solve each part of ITMP separately. Existing task planning algorithms include heuristic search [1], constrained-based method [2], logic programming [3], etc. Correspondingly, the motion planning algorithms are mostly based on heuristic search [4], sampling [5], [6], and numerical optimization [7], [8]. However, when it comes to the integration of both task and motion planning, it is challenging to optimize task level and motion level decisions simultaneously.

Hierarchical planning approaches have been developed to solve ITMP problems. For example, Dornhege *et al.* [9] proposes a task planner steering the search, while the motion planner is called to check feasibility via semantic attachments. Wolfe *et al.* [10] encodes the robotic manipulation problems as vertically hierarchical task networks and presents the state-abstracted hierarchical task network algorithm to obtain a feasibility guaranteed kinematic plan. Srivastava *et al.* [11] integrates the off-the-shelf task planner and motion planner and a heuristic method is used to remove potentially obstructing objects. However, the employment of exhaustive search is time-consuming and thus has limited scalability. To expedite the computational speed, Kaelbling and Lozano-Pérez[12] interweaves planning and execution to reduce the search space, while reversible actions are required for possible backtracking. The other planning approaches based on hierarchical frameworks can be found in [13]–[15]. Additionally, approaches for constraint satisfaction problem, Lozano-Pérez and Kaelbling[16] formulate the ITMP in a discretized configuration space and perform a high-level symbolic search. This approach is also applied to deal with ITMP problems with geometric constraints [17], [18]. Other ITMP approaches include fast forward planner [19] and incremental framework [20], among others. In summary, these approaches for ITMP generally require symbolic goal description and focus on finding a feasible solution instead of an optimal one.

When aiming to optimize a desired performance index of robotic systems, we propose to solve the ITMP through an optimization lens. Work in [21] puts forward a multilevel optimization approach to solve the ITMP. In addition, a logic-geometric programming approach is presented in [22] with the cost function defined by the terminal states and the constraints defined by the symbolic state-action sequence. However, this approach is computationally inefficient when the scale of involved tasks increases. Similar to the problem considered in this article, Kiesel *et al.* [23] solves the TSP in a multilevel optimization framework combing tabu search, linear programming, and heuristic search. Genetic algorithm (GA) has been applied to solve the ITMP for ground vehicles [24]. However, like most heuristic methods, GA requires a great amount of computational efforts and thus is not applicable to large-scale problems. Furthermore, Cons *et al.* [25] introduces a modified TSP by integrating the vehicular constraints using a Dubins type curve with varying turning rates. After reviewing the existing algorithms for ITMP, there arises the necessity to develop an advanced optimization method with high performance (optimality), robust convergence, and improved scalability.

Hybrid systems arise in many fields, such as communication networks, air transportation management, manufacturing, just to name a few. A comprehensive survey of modeling, analysis, and control of hybrid systems can be found in [26]. Such a system of interest typically is composed of at least two types of dynamical behaviors interacting with each other. Analysis and control algorithms for hybrid systems can be referred to [26]–[29] and the references therein. Similar to ITPM, optimal control of hybrid systems is characterized by mixed-type decision variables representing both dynamical modes and system states. One typical hybrid control problem presented here is the rendezvous and docking of a robotic charging station, where two dynamical modes will switch at a point defined by the relative distance between the charging station and the approaching robot.

Mobile robot charging can date back to 1950s [30], where light is used to find the way to the charging station. A stationary charging station with a docking mechanism for a mobile robot has been developed by Silverman *et al.* [31]. After that, Cassinis *et al.* [32]–[34] develop the hardware as well as control strategies of docking and charging systems for a single robot. Additionally, Drenner and Papanikolopoulos[35] addresses a similar problem to optimize the locations of mobile docking stations to maximize the available power of a deployed robot team. More recently, reinforcement learning technique has been applied to the docking and charging for a humanoid robot [36]. Multirobot rendezvous planning for recharging has also been investigated in persistent tasks [37].

After formulating the class of mixed-type decision-making problems as nonconvex QCQPs, we focus on developing an optimization algorithm to efficiently solve QCQPs. A nonconvex QCQP is equivalent to a linear matrix programming problem by introducing a to-be-determined rank-one matrix [38]. Extensive relaxation methods have been developed to find a lower bound on the objective value of a QCQP, which cannot guarantee yielding a feasible solution [39]. Approaches based on approximation and iterations depend on a good initial guess and fast convergence is not guaranteed [40], [41]. Recent work by Sojoudi and Lavaei [42] proposes that the structure of the QCQP problems can be changed based on graph theory to obtain a low-rank solution, which greatly reduces the gap between the exact solution and the relaxed one. Furthermore, work by Lasserre [43] generates a series of semidefinite programs to solve polynomial optimization problems, which is applicable to small-scale QCQPs.

ADMM is a well-known sequential algorithm that divides the unknown variables into two sets and solves them separately in an alternative sequence when solving both sets jointly is more complicated [44]. The fundamental principle of ADMM is to formulate the augmented Lagrangian and then find the first-order conditions for the two unknown sets in an alternative sequence, followed by updating the Lagrangian multipliers associated with equality constraints. Rigorous convergence property has been proved for ADMM in solving convex optimization problems [45]. However, there is a general lack of convergence justification for ADMM when solving nonconvex optimization problems. Most existing works of ADMM for nonconvex optimization focus on problems with nonconvexity in the objective function only [46], [47] or matrix factorization [48], [49]. Under a variety of assumptions, e.g., convergence or boundedness of dual objectives, they have demonstrated converge to a Karush–Kuhn–Tucker (KKT) stationary point for these special nonconvex problems.

After examining the existing literature, we find that only a few works deal with nonconvex constraints when applying the ADMM framework. For example, Jiang *et al.* [50] tackles polynomial optimization problems by minimizing a general objective over a spherical constraint $\|x\|_2 = 1$ and Shen *et al.* [51] solves the low-rank and sparse matrix separation problem. General QCQPs solved by ADMM have been investigated by Huang and Sidiropoulos [52]. However, every subproblem in the existing ADMM framework still requires to be solved iteratively in most cases, which generates extra computational burden. Moreover, the convergence analysis is incomplete, which only proves that all limit points are also KKT stationary points without proof of convergence to the limit points through the iterations. Other researchers have investigated the convergence property of ADMM for some special instances [53]. This article makes significant additions to our previous work in [54] by extending a specific hybrid control problem to general mixed-type decision-making problems. Furthermore, rigorous convergence proof of the proposed optimization algorithm is provided. More importantly, the unified formulation and algorithm are implemented and verified in experimental tests.

B. Contribution

The contribution of this article for mixed-type decision-making is in twofolds, including the problem formulation and a novel optimization algorithm. The unified representation in form of QCQPs avoids the efforts required to solve a wide range of mixed-type decision-making problems individually. Furthermore, the contributions of the proposed algorithm for solving general QCQPs also include two aspects. First, the algorithm will always admit efficient and closed-form solutions based on simple linear matrices operations for all subproblems, which indicates that it does not require any optimization solvers in the iterative algorithm and can hence greatly improve the efficacy and implementability. It is also worth noting that such property applies to QCQPs with both equality and inequality constraints. Second, the conditional convergence property is strengthened compared to the related literature. Under certain mild assumptions, we prove the global convergence of the proposed nonconvex optimization method. In addition to virtual simulation, we verify effectiveness of the performance value for a designated objective function, efficacy in terms of computation time, and robustness indicating insensitive to the initial guess when implementing in real-world experimental tests.

The rest of this article is organized as follows. We first present the formulation of general mixed-type decision-making problems followed by the transformation of two representative problems into QCQPs. Next, the ADMM framework for nonconvex QCQP and the solutions of subproblems are derived with the convergence analysis of the proposed algorithm. The comparative simulation and experimental results are then presented. We conclude this article with a few remarks at the end.

C. Preliminaries

The notation used throughout this article is introduced in this section. The set of $n \times n$ symmetric matrices is denoted by $\mathbb{S}^n$ and the set of $n \times n$ positive semidefinite (definite) matrices is denoted by $\mathbb{S}^n_+$ ($\mathbb{S}^n_{++}$). The notation $X \succeq \mathbf{0}$ ($X \succ \mathbf{0}$) designates that the matrix $X \in \mathbb{S}^n$ is positive semidefinite (definite) and I denotes an identity matrix in an appropriate dimension. The Frobenius norm of X is denoted by $\|X\|_F$. The operator $\text{Proj}_{\mathcal{C}}(X_0) = \arg\min_{X \in \mathcal{C}} \|X - X_0\|$ represents the projection of X_0 onto the set $\mathcal{C}$. For X_1 and X_2 in the same dimension, $\langle X_1, X_2 \rangle$ represents the inner product. For a number x, $\lfloor x \rfloor$ denotes the floor of x. For a set $\mathcal{E}$, $|\mathcal{E}|$ denotes its cardinality. For a vector x, $|x|$ denotes its elements-wise absolute value. For a matrix X and vector x in appropriate dimensions, $X \setminus x$ denotes the left division. For a matrix X and vector x in appropriate dimensions, $x = \text{vec}(X)$ is the vectorization by stacking one column of X onto the other in order. Correspondingly, $X = \text{mat}(x)$ is the reverse. $X = \mathbf{Diag}(x)$ denotes a diagonal matrix with x as the diagonal entries.

II. Problem Formulation and Conversion

In this section, formulation of the class of mixed-type decision-making problems is introduced first, followed by the description and conversion of the two representative problems.

A. General Mixed-Type Decision-Making Problem

We consider the type of mixed-type decision-making problems with polynomial objective and constraint set, which can be generalized as mixed-variable polynomial optimization problems, expressed as

$$\begin{aligned} J &= \min_{\mathbf{z}} p(\mathbf{z}) \\ \text{s.t. } & q_j(\mathbf{z}) \leq 0,\ j = 1, \ldots, m \\ & \mathbf{z}^d \in \{0, 1\},\ d \in D \end{aligned} \tag{2.1}$$

where $\mathbf{z}$ represents the decision variable set, D is the index set for binary decision variables, the objective J is a polynomial function. For system dynamics in form of $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ with $\mathbf{x}, \mathbf{u}$ representing state and control variables, respectively, constraints from dynamics could be transformed into polynomial equalities through discretization when $\mathbf{f}(\mathbf{x}, \mathbf{u})$ are original expressed as polynomials. Together with operational and mission constraints, the general polynomial equalities/inequalities in form of $q_j(\mathbf{z}) \leq 0,\ j = 1, \ldots, m$, are used to represent all constraints for the entire decision-making problem. To unify the type of decision variables, a discrete decision value, $z \in \{0, 1\}$, can be regarded as a continuous variable with a quadratic constraint, $z(z - 1) = 0$.

Every polynomial function can be equivalently transformed into a quadratic function by introducing additional variables and quadratic equality constraints [55]. For example, $g(a, b) = ab^2$ can be equivalently represented by a quadratic function $g(a, c) = ac$ with a quadratic equality $c = b^2$. For problems with trigonometric terms, e.g., $\sin\alpha$ and $\cos\alpha$ appearing in the formulation, these trigonometric terms can be replaced by new variables $x_1 = \sin\alpha$ and $x_2 = \cos\alpha$ with an additional quadratic constraint $x_1^2 + x_2^2 = 1$. Through quadratic conversions of the binary variable set, $z^d \in \{0, 1\},\ d \in D$, together with the quadratic representations for polynomial constraints and objective function, the class of mixed-type decision-making problems can be cast as a general/nonconvex QCQP in form of

$$\begin{aligned} \min_{\tilde{x} \in \mathbb{R}^n} \ & \tilde{x}^T \tilde{A}_0 \tilde{x} + \tilde{a}_0^T \tilde{x} \\ \text{s.t. } & \tilde{x}^T \tilde{A}_i \tilde{x} + \tilde{a}_i{}^T \tilde{x} = c_i, i \in \mathcal{E} \\ & \tilde{x}^T \tilde{B}_j \tilde{x} + \tilde{b}_j^T \tilde{x} \leq d_j, j \in \mathcal{I} \end{aligned} \tag{2.2}$$

where $\tilde{A}_0,\ \tilde{A}_i\ (i \in \mathcal{E}),\ \tilde{B}_j\ (j \in \mathcal{I}) \in \mathbb{S}^n$ and $\tilde{a}_0,\ \tilde{a}_i\ (i \in \mathcal{E}),\ \tilde{b}_j\ (j \in \mathcal{I}) \in \mathbb{R}^n$ with $\mathcal{E}$ and $\mathcal{I}$ denoted as the indices set of the equality and inequality constraints, respectively. In the following, we will use two representative mixed-type decision-making problems to explain the conversions from the original problem formulation into a QCQP. One represent integrated task and mission planning with system dynamics/constraints expressed as polynomial and/or trigonometric functions. The other one represent the type of hybrid optimal control problems where system dynamics and constraints are expressed as polynomial and/or trigonometric functions. Moreover, the switching conditions of dynamics/control/constraints must be expressed as polynomials as well.

B. Traveling Salesman Problem With Obstacle Avoidance and Unicycle Kinematics

TSP is to determine the visiting sequence and paths between two adjacent waypoints for a UGV with an assigned starting point and a group of visiting waypoints. Considering the real-world operational restrictions, avoidance zones, where the UGV cannot travel through, are included in the problem as constraints. In the following context, we will focus on modeling the UGV kinematics in the two-dimensional space, which can be easily extended to three-dimensions. The kinematics of UGV in two-dimensions is represented by a unicycle model in the form of

$$\begin{aligned} \dot{x} &= V\cos\theta \\ \dot{y} &= V\sin\theta \\ \dot{\theta} &= u \\ |u| &\leq u_{\max} \end{aligned} \tag{2.3}$$

where x and y are the coordinates, V is the specified speed, θ is the heading angle, and $u_{\max}$ is the maximum heading angle changing rate. (x, y, θ) are shorthand for their respective trajectories $(x(\bullet), y(\bullet), \theta(\bullet))$ which are functions of time. The starting point is specified as $[x_0, y_0]$. The feasible paths considering the obstacle avoidance constraints are formulated as

$$\{(x, y) | \mathbf{g}_A(x, y) \geq 0\} \tag{2.4}$$

where $\mathbf{g}_A : f \to \mathbb{R}^{n_A}$ is a functional and n_A is the number of the avoidance zones, (x, y) represent coordinates of the feasible paths satisfying the obstacle avoidance constraints. For different geometric shapes of the avoidance zones, $\mathbf{g}_A$ has corresponding expressions. Here, we focus on elliptical avoidance zones (with circles as special cases), expressed as $(\mathbf{g}_A)_i = (\frac{x - x_i^e}{a_i^e})^2 + (\frac{y - y_i^e}{b_i^e})^2 - 1$ with (x_i^e, y_i^e) denoted as the center of the ellipse and a_i^e, b_i^e denoted as the semimajor and semiminor axes of the ellipse, respectively, for $i = 1, \ldots, n_A$. Moreover, the waypoints required to be visited are denoted by $(x_p^w, y_p^w),\ p = 1, \ldots, P$, with P as the total number of waypoints.

For a minimum time path, the performance index is determined by $J = \int_{t_0}^{t_f} 1dt$. This is identical to the minimum distance index providing a constant speed. Alternative objective functions, such as the minimum control efforts, can be easily adapted here. Consequently, the minimum time TSP for a UGV with unicycle kinematics and obstacle avoidance can be formulated as

$$\begin{aligned} \min_u \ & \int_{t_0}^{t_f} \ 1dt \\ \text{s.t. } & x(t_0) = x_0, y(t_0) = y_0 \\ & \dot{x} = V\cos\theta \\ & \dot{y} = V\sin\theta \\ & \dot{\theta} = u \\ & |u| \leq u_{\max} \\ & \mathbf{g}_A(x, y) \geq 0 \\ & (x_p^w, y_p^w) \in (x, y), p = 1, \ldots, P \end{aligned} \tag{2.5}$$

where the last two constraints represent the obstacle avoidance and the waypoints-visiting constraints, respectively. The above decision-making problem is a nonlinear optimization problem, including differential kinematics and nonconvex constraints on

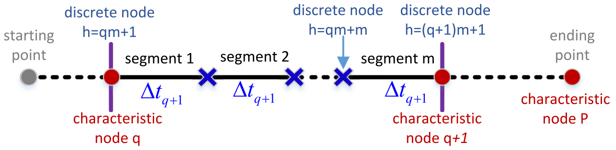


Fig. 1. Illustration of the discretization scheme.

the states. The last set of logical constraints, which is uncommon in the classical optimal control problems, makes it even more challenging. As a result, we aim to discretize the above mixed-type decision-making problem and solve it as a parameter optimization problem.

C. Reformulation of the Traveling Salesman Problem as a QCQP Problem

Formulation of the TSP with unicycle kinematics is different from the traditional modeling approach where a straight line is assumed between two connected waypoints and adjacency matrix is employed to represent the connectivity among waypoints. Due to the involved mixed-type decision variables and nonlinear/nonconvex constraints, a new modeling approach is proposed to reduce computational complexity. The unicycle model described in (2.3) for the UGV path planning includes trigonometric functions, which are highly nonlinear. The first step toward solving the TSP with avoidance zones is to discretize the continuous path and formulate the problem as a parameter optimization problem. As a result, the continuous path is discretized into a series of nodes with coordinates $[x_h, y_h]$, $h = 1, \ldots, H$, where H is the total number of discrete nodes.

In the following, we first reformulate the waypoints-visiting constraints as quadratic and linear constraints. Recall that locations of waypoints $(x_p^w, y_p^w), p = 1, \ldots, P$, are given, we construct a new subset of P nodes, named *characteristic nodes*, where index of each node, $q = 1, \ldots, P$ is given and node q is connected only with node $q+1$, for all $q = 1, \ldots, P-1$, but locations of these nodes are unknown. A waypoint (x_p^w, y_p^w), $p = 1, \ldots, P$, will coincide with one and only one characteristic node. The correspondences between the paired characteristic node and the waypoint are to be determined, which is an alternative approach to determine the visiting sequences. Advantages of the new modeling approach in terms of reducing computational complexity will be addressed later.

Between each pair of connected characteristic nodes q and $q+1$, for all $q = 1, \ldots, P-1$, the path between them is discretized into m segments with indices of discretization nodes, denoted by $h = qm+1, \ldots, (q+1)m$ and the first one with index $h = qm+1$ is assumed to be the characteristic node q. The given starting point, denoted by $q = 0$, is connected with characteristic node $q = 1$ and discretized into m segments as well. Thus, the overall number of discrete nodes is $H = Pm$. The discretization scheme is illustrated in Fig. 1, where the time duration of each segment between two adjacent characteristic nodes q and $q+1$ is assumed to be identical, denoted by Δt_{q+1}, but may be different from the segment duration between another pair of adjacent characteristic nodes. These settings significantly reduce the complexity without losing any degrees of freedom in the decision-making. Mathematically, the waypoints-visiting constraints are formulated as

$$\begin{aligned}
&\Gamma_{pq} \geq \frac{(x_{qm+1} - x_p^w)^2 + (y_{qm+1} - y_p^w)^2}{M^2} \\
&\forall\, p, q = 1, \ldots, P \\
&\sum_{p=1}^{P} \Gamma_{pq} = P - 1, \forall\, q = 1, \ldots, P \\
&\Gamma \in \{0, 1\}^{P \times P}
\end{aligned} \tag{2.6}$$

where $M \in \mathbb{R}_{++}$ is a predefined sufficiently large constant. It could be the maximum distance between any pair of the nodes within the designated area. Then, $\Gamma_{pq} = 0$ indicates that the qth characteristic node in the path coincides with the pth waypoints in the task level. Moreover, the column-sum constraint on $\Gamma \in \{0, 1\}^{P \times P}$ is to guarantee that each waypoint is visited by one and only one characteristic node. Additionally, given that one characteristic node can only coincide with one waypoint (constrained by the inequality in (2.6) under the assumption that waypoints do not overlap with each other) and the number of characteristic nodes and waypoints are the same, it further implies that such coincidence occurs at a one-by-one basis between the characteristic nodes and waypoints. In other words, there is one and only one zero element in each row or column of Γ, though it is only explicitly enforced column-wisely.

The new formulation of the visiting sequence constraints generates additional P^2 discrete variables. Instead of having those characteristic nodes with known indices designated to coincide with the waypoints, there is a more general but resource-consuming formulation approach. The traditional formulation assumes that every discrete node, $h = 1, \ldots, H$, in the path can possibly coincide with one of the waypoints. In parallel, this setting will produce HP discrete variables. Moreover, for the first inequality in (2.6), the number of inequalities turns out to be HP instead of P^2. As in a real-world TSP with several visiting waypoints, it usually holds that $H = Pm \gg P$. The much fewer introduced variables together with a reduced number of constraints validate the computational advantages of the proposed discretization approach.

In the following, we will show that the obstacle avoidance constraints can also be converted into the quadratic form. As an elliptical zone is defined by a single second-order equation, the conversion of obstacle avoidance constraints from the continuous path to the discretized one is straightforward. That is, the discretized nodes can be directly plugged into the elliptical avoidance constraint, which are expressed as:

$$\begin{aligned}
&\left(\frac{x_h - x_i^e}{a_i^e}\right)^2 + \left(\frac{y_h - y_i^e}{b_i^e}\right)^2 - 1 \geq 0 \\
&\forall h = 1, \ldots, H,\ i = 1, \ldots, n_A
\end{aligned} \tag{2.7}$$

where $i = 1, \ldots, n_A$ are the indices of the elliptical zones.

At last, it comes to the reformulation of the kinematic differential equation. The differentials can be expressed in the discretized form as follows:

$$\dot{x} = \frac{x_{h+1} - x_h}{\Delta t_{\lfloor (h-1)/m \rfloor + 1}} = V\cos\theta_h,\ h = 1,\ldots,H-1 \tag{2.8}$$

$$\dot{y} = \frac{y_{h+1} - y_h}{\Delta t_{\lfloor (h-1)/m \rfloor + 1}} = V\sin\theta_h,\ h = 1,\ldots,H-1 \tag{2.9}$$

where $\Delta t_{\lfloor (h-1)/m \rfloor + 1} = \Delta t_{q+1}$ is the time interval between node h and node $h+1$ and it is the same for all segments between two adjacent characteristic nodes q and $q+1$ as stated above. In the following context, $\Delta t_{(h)}$ is used to denote $\Delta t_{\lfloor (h-1)/m \rfloor + 1}$ for notational ease. The above two equations can be synthesized as

$$(x_{h+1} - x_h)^2 + (y_{h+1} - y_h)^2 = V^2(\Delta t_{(h)})^2$$
$$\forall\, h = 1,\ldots,H-1. \tag{2.10}$$

Differentiating (2.8)–(2.9) leads to

$$\ddot{x} = \tfrac{x_{h+2} + x_h - 2x_{h+1}}{(\Delta t_{(h)})^2} = -V\dot{\theta}_h\sin\theta_h,\ \forall h = 1,\ldots,H-2$$

$$\ddot{y} = \tfrac{y_{h+2} + y_h - 2y_{h+1}}{(\Delta t_{(h)})^2} = V\dot{\theta}_h\cos\theta_h,\ \forall h = 1,\ldots,H-2.$$

Combining with $\dot{\theta} = u$ and $|u| \le u_{\max}$, the above equations can be synthesized as

$$\dot{\theta}_h^2 = \left[\frac{x_{h+2} + x_h - 2x_{h+1}}{V\Delta t_{(h)}^2}\right]^2 + \left[\frac{y_{h+2} + y_h - 2y_{h+1}}{V\Delta t_{(h)}^2}\right]^2$$
$$\le u_{\max}^2. \tag{2.11}$$

By introducing an additional variable $t'_{(h)} = \Delta t_{(h)}^2$, the above constraint can be reformulated as a quadratic inequality in the form of

$$(x_{h+2} + x_h - 2x_{h+1})^2 + (y_{h+2} + y_h - 2y_{h+1})^2$$
$$\le V^2 u_{\max}^2 {t'_{(h)}}^2, \forall\, h = 1,\ldots,H-2. \tag{2.12}$$

So far, by discretization, the differential kinematic equations involving trigonometric functions are transformed into quadratic constraints, while the rest, i.e., the objective and initial conditions, are linear.

A binary variable $\Gamma_{pq} \in \{0,1\}$ can be regarded as a continuous variable with a quadratic constraint, i.e., $\Gamma_{pq}(\Gamma_{pq} - 1) = 0$. Then, the TSP with avoidance zones is converted into a continuous optimization problem with quadratic and linear constraints, which is generalized as a nonconvex QCQP problem in the form of

$$\min_{x,y,\Delta t,t,'\gamma} \quad m\sum_{q=1}^{P} \Delta t_q$$
$$\text{s.t.}\quad x(t_0) = x_0,\ y(t_0) = y_0$$
$$t'_q = (\Delta t_q)^2, \forall\, q = 1,\ldots,P$$
$$\text{(2.6), (2.7), (2.10), (2.12).} \tag{2.13}$$

With a given number of discrete nodes, this minimum time problem will minimize Δt_q while satisfying all specified constraints. Given a sufficiently large P, the optimization will lead to a fine discretized trajectory close to the true one. With quadratic equality and inequality constraints and associated nonpositive semidefinite Hessian matrices, problem in (2.13) is nonconvex and classified as NP-hard.

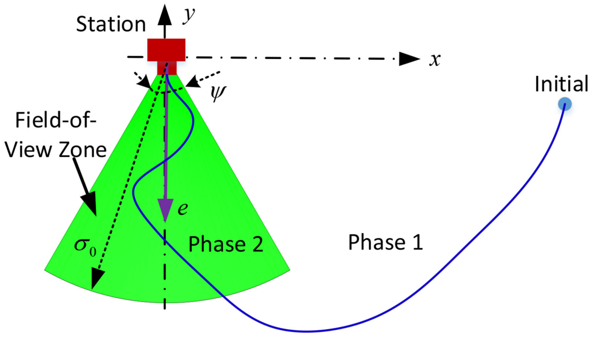


Fig. 2. Illustration of the multiphase mission (not to scale).

D. Robot Rendezvous and Docking Problem

The second mixed-type decision-making problem we consider is the multiphase mission planning problem where a UGV with given initial states is designated to accomplish the rendezvous and docking task at a stationary charging station. This problem has practical applications in robot refueling or recharging missions. The rendezvous and docking maneuver is composed of two phases defined by the relative distance between the UGV and the station, shown in Fig. 2, where phase two has more restrictive constraints. The field-of-view zone in phase two is a two-dimensional cone centered at the target and defined by radius σ_0, constrained angle ψ, and boresight vector $\mathbf{e}$. For notational convenience, the target is set at the origin, i.e., (0,0) and $\mathbf{e} = [0,-1]$. The UGV travels at a constant speed V_1 and V_2 at phases one and two, respectively, with unicycle kinematics. To achieve high docking precision and reduce impacts to the charging station at the end of phase two, it assumes that V_2 is much smaller than V_1. Moreover, in phase two, the UGV is kept within the field-of-view zone of the station. Consequently, the challenge to formulate such a hybrid control problem is to explicitly express the entire problem with distinct constraints of two phases in a unified form. To address this challenge, the planned path is again discretized into H nodes and binary variables associated with each discretization node are introduced. For node h, $h = 1,\ldots,H$, $a_h \in \{0,1\}$ is defined as

$$a_h = \begin{cases} 0, & \text{node } h \text{ is in phase one} \\ 1, & \text{node } h \text{ is in phase two.} \end{cases} \tag{2.14}$$

As a result, we have a unified representation of the speed as follows:

$$V_h = a_h(V_2 - V_1) + V_1.$$

The UGV considered here has the same kinematics as expressed in (2.3). However, unlike the TSP, a unique time interval Δt between each pair of adjacent nodes is assumed for this hybrid control problem. As a result, the same discretization scheme stated in (2.10) and (2.12) is used to convert the kinematics into quadratic constraints, expressed as

$$\begin{aligned}&(x_{h+1}-x_h)^2+(y_{h+1}-y_h)^2 = V_h^2 t,' \ \forall\, h = 1,\ldots,H-1\\ &(x_{h+2}+x_h-2x_{h+1})^2+(y_{h+2}+y_h-2y_{h+1})^2\\ &\le V_h^2 u_{\max}^2 t,'' \ \forall\, h = 1,\ldots,H-2 \qquad (2.15)\end{aligned}$$

with $t' = (\Delta t)^2$ and $t'' = (t')^2$. Note that though V_h^2, $h = 1,\ldots,H-2$, is in the quadratic form, it is linearly determined by the unknown variable a_h that is constrained by $a_h^2 = a_h$. Then, we have

$$\begin{aligned}V_h^2 &= a_h^2(V_2-V_1)^2+2(V_2-V_1)V_1a_h+V_1^2\\ &= (V_2^2-V_1^2)a_h+V_1^2.\end{aligned}$$

As a result, the multiphase kinematics in (2.15) is quadratic. Furthermore, the field-of-view constraint in a cone for phase two is expressed as

$$\begin{aligned}&\sigma_h \le \sigma_0, \forall \text{ node } h \text{ is in phase two}\\ &\frac{(x_h,y_h)\bullet \mathbf{e}}{\sigma_h\|\mathbf{e}\|} \ge \cos(\frac{\psi}{2}), \forall \text{ node } h \text{ in phase two} \qquad (2.16)\end{aligned}$$

where $\sigma_h = \sqrt{x_h^2+y_h^2}$ represents the distance between node h and the station. Intuitively, the first inequality restricts the radial distance while the second one restricts the angular offset with regard to the boresight vector of the cone. Without loss of generality, assuming that direction of the negative y axis coincides with vector $\mathbf{e}$, then $\mathbf{e} = [0,-1]$ and the second constraint in (2.16) can be further simplified as

$$\sigma_h \cos\left(\frac{\psi}{2}\right)+y_h \le 0, \forall \text{ node } h \text{ in phase two.} \qquad (2.17)$$

Moreover, the first phase is constrained to be out of the cone and also outside the circle with radius σ_0 in order to make the transition to the second phase smooth. Consequently, we come to the following constraint

$$\sigma_h \ge \sigma_0, \forall \text{ node } h \text{ in phase one.} \qquad (2.18)$$

Since constraints in (2.16) and (2.17) are active only in phase two, conversely, constraints in (2.18) are only active in phase one, the binary variables a_h, $h = 1,\ldots,H$, defined in (2.14) are used to rewrite constraints (2.16)–(2.18) in a unified quadratic form as follows:

$$\begin{aligned}&(1-a_h)(-\sigma_h+\sigma_0) \le 0\\ &a_h(\sigma_h-\sigma_0) \le 0\\ &a_h\left(\sigma_h\cos\left(\frac{\psi}{2}\right)+y_h\right) \le 0. \qquad (2.19)\end{aligned}$$

Additional constraints to prevent abnormal operations are considered and expressed as

$$\begin{aligned}&a_{h+1} \ge a_h,\ h = 1,\ldots,H-1\\ &(a_{h+1}-a_h)(\sigma_{h+1}-\sigma_0) = 0,\ h = 1,\ldots,H-1 \qquad (2.20)\end{aligned}$$

where the first constraint guarantees that the UGV will remain within the cone once it enters, the second one guarantees that the entering point into the cone is on the arc of the field-of-view zone. Accordingly, the UGV rendezvous and docking problem is summarized as a nonconvex QCQP problem in the following:

$$\begin{aligned}\min_{x,y,\Delta t,t,'t,''a} \quad & (H-1)\Delta t\\ \text{s.t.} \quad & x(t_0) = x_0, y(t_0) = y_0\\ & t' = (\Delta t)^2, t'' = (t')^2\\ & (2.15)\\ & (2.19)\\ & (2.20). \qquad (2.21)\end{aligned}$$

Similar to the TSP described above, the mixed-type decision-making problem in (2.21) is nonconvex and NP-hard.

III. Algorithm: Framework and Procedures

Both representative problems described above illustrate how a mixed-type decision-making problem of robotic systems is formulated as a general/nonconvex QCQP. In fact, a wide range of these problems with nonlinear dynamics and/or operational/mission constraints can be generalized as a nonconvex QCQP presented in (2.2). The general form of QCQP in (2.2) with inhomogeneous quadratic function can be transformed into homogeneous ones by introducing a new variable $\beta \in \mathbb{R}$ and a new quadratic constraint $\beta^2 = 1$ by the following formulation:

$$\begin{aligned}\min_{\tilde x,\beta} \quad & \begin{bmatrix}\tilde x^T & \beta\end{bmatrix}\begin{bmatrix}\tilde A_0 & \tilde{a_0}/2\\ \tilde{a_0}^T/2 & 0\end{bmatrix}\begin{bmatrix}\tilde x\\ \beta\end{bmatrix}\\ \text{s.t.} \quad & \begin{bmatrix}\tilde x^T & \beta\end{bmatrix}\begin{bmatrix}\tilde A_i & \tilde{a_i}/2\\ \tilde{a_i}^T/2 & 0\end{bmatrix}\begin{bmatrix}\tilde x\\ \beta\end{bmatrix} = c_j, i \in \mathcal{E}\\ & \begin{bmatrix}\tilde x^T & \beta\end{bmatrix}\begin{bmatrix}\tilde B_j & \tilde{b_i}/2\\ \tilde{b_i}^T/2 & 0\end{bmatrix}\begin{bmatrix}\tilde x\\ \beta\end{bmatrix} \le d_j, j \in \mathcal{I}\\ & \beta^2 = 1. \qquad (3.22)\end{aligned}$$

Then x^*/β^* will be the solution to the original problem stated in (2.2) while (x^*,β^*) is the solution pair of (3.22). In addition, linear constraints in (2.13) can be rewritten in the above quadratic form as well by setting the corresponding Hessian as a zero matrix. Therefore, the general homogeneous QCQP can be formulated as

$$\begin{aligned}\min_{x\in\mathbb{R}^n} \quad & x^TA_0x\\ \text{s.t.} \quad & x^TA_ix = c_i, i \in \mathcal{E}\\ & x^TB_jx \le d_j, j \in \mathcal{I}. \qquad (3.23)\end{aligned}$$

By introducing the slack variables $s \in \mathbb{R}^{|\mathcal{I}|}$, problem in (3.23) is equivalent to

$$\begin{aligned}\min_{x} \quad & x^TA_0x\\ \text{s.t.} \quad & x^TA_ix = c_i, i \in \mathcal{E}\\ & x^TB_ix + s_j = d_j, j \in \mathcal{I}\\ & s \ge 0. \qquad (3.24)\end{aligned}$$

Furthermore, auxiliary variable, $y \in \mathbb{R}^n$, is introduced to transform the objective as well as the constraints to be bilinear

$$\begin{aligned}\min_{x,y,s} \quad & x^T A_0 y \\ \text{s.t.} \quad & x^T A_i y = c_i, i \in \mathcal{E} \\ & x^T B_i y + s_j = d_j, j \in \mathcal{I} \\ & s \geq 0. \\ & x = y. \end{aligned} \tag{3.25}$$

Then, the augmented Lagrangian function of (3.25) is constructed as

$$\begin{aligned}\mathcal{L}(x, y, s, \mu, \lambda, \nu) &= x^T A_0 y + \langle \mu, x^T A_{\mathcal{E}} y - c \rangle \\ &+ \langle \lambda, x^T B_{\mathcal{I}} y + s - d \rangle + \langle \nu, x - y \rangle + \frac{\rho_1}{2} \| x^T A_{\mathcal{E}} y - c \|_F^2 \\ &+ \frac{\rho_2}{2} \| x^T B_{\mathcal{I}} y + s - d \|_F^2 + \frac{\rho_3}{2} \| x - y \|_F^2 \end{aligned} \tag{3.26}$$

where $\mu \in \mathbb{R}^{|\mathcal{E}|}, \lambda \in \mathbb{R}^{|\mathcal{I}|}, \nu \in \mathbb{R}^n$ are the dual variables associated with the corresponding coupling constraints. For notational convenience, we denote $x^T A_{\mathcal{E}} y - c \in \mathbb{R}^{|\mathcal{E}|} := [x^T A_1 y - c_1, \ldots, x^T A_{|\mathcal{E}|} y - c_{|\mathcal{E}|}]^T$, and similarly for $x^T B_{\mathcal{I}} y + s - d \in \mathbb{R}^{|\mathcal{I}|}$. Moreover, $\rho_1 > 0, \rho_2 > 0$, and $\rho_3 > 0$ are the penalty coefficients for the augmented terms, respectively. In the following, all the matrix norms are Frobenius norm unless explicitly indicated. By introducing the auxiliary variable set y and a new constraint $x = y$, the quadratic terms in (3.24) turn to be bilinear in (3.25). Then, the augmented terms in the Lagrangian becomes quadratic with respect to variables in every block. Without introducing the auxiliary variable set y, the Lagrangian of the original problem will have quartic augmented terms, which is not necessarily (strongly) convex. Even they are convex, solving the first-order optimality condition expressed as third-order polynomials is challenging. These reasons justify the advantages of introducing the auxiliary variable set y in (3.25).

By employing the classical ADMM framework, we partition the variables into three sets, x, y, and s. To solve (3.25), the algorithm solves the three variable sets in sequence, which is followed by the update of the dual variables (μ, λ, ν). The subproblems of ADMM at step $r + 1$ for (3.25) include

$$x^{r+1} := \arg\min_{x} \mathcal{L}(x, y^r, s^r; D^r) \tag{3.27a}$$

$$y^{r+1} := \arg\min_{y} \mathcal{L}(x^{r+1}, y, s^r; D^r) \tag{3.27b}$$

$$s^{r+1} := \arg\min_{s \geq 0} \mathcal{L}(x^{r+1}, y^{r+1}, s; D^r) \tag{3.27c}$$

$$\begin{aligned} \mu^{r+1} &:= \mu^r + \rho_1^r ((x^{r+1})^T A_{\mathcal{E}} y^{r+1} - c) \\ \lambda^{r+1} &:= \lambda^r + \rho_2^r ((x^{r+1})^T B_{\mathcal{I}} y^{r+1} + s^{r+1} - d) \\ \nu^{r+1} &:= \nu^r + \rho_3^r (x^{r+1} - y^{r+1}) \end{aligned} \tag{3.27d}$$

where we denote $D^k = (\mu, \lambda, \nu)^k$ as a collection of the dual variables here and in the following context for notational ease.

A. Solutions to the Subproblems

While the updates of the dual variables in (3.27d) are trivial, the sequential updates of primal variables require to solve convex optimization problems. From the subproblems constructed in (3.27a)–(3.27c), they imply that all the three updates of primal are quadratic programming with the first two unconstrained and the third one constrained by nonnegativity constraints. Note that all three optimization problems are all strongly convex so that the (global) optima are well defined. We first explain how to solve the x-update problem. The first-order optimality condition of (3.27a) is

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial x} &= A_0 y + \sum_{i=1}^{|\mathcal{E}|} \mu_i A_i y + \sum_{j=1}^{|\mathcal{I}|} \lambda_j B_j y + \nu \\ &+ \rho_1 \sum_{i \in \mathcal{E}} A_i y \left[(A_i y)^T x - c_i \right] \\ &+ \rho_2 \sum_{j \in \mathcal{I}} B_j y \left[(B_j y)^T x + s_j - d_i \right] \\ &+ \rho_3 (x - y) = 0 \end{aligned} \tag{3.28}$$

where the superscript "r" denoting the ADMM updating sequence is omitted here. Solving (3.28) will give the global optimum of (3.27a). Then, the backslash operator with $O(n^2)$ complexity, instead of using the matrix inversion with up to $O(n^3)$ complexity, is exploited here to solve the linear equality in (3.28). In addition, here and in quite a number of optimization problems derived from discretization, the parameter matrices $A_0, A_i, i \in \mathcal{E}$ and $B_j, j \in \mathcal{I}$ are usually sparse. As a result, fully exploiting such sparsity will greatly reduce the complexity of operations, including the matrix multiplication and the left division. Moreover, to promote linear algebra efficiency when computing matrix H_x and vector b_x during the update of x in (3.27a), constructing computationally efficient matrices structures are considered. For example, $\sum_{i \in \mathcal{E}} (A_i y)(A_i y)^T = A_y A_y^T$, where $A_y \in \mathbb{R}^{n \times |\mathcal{E}|} = \text{mat}(\hat{A} y)$ and $\hat{A} = [A_1, \ldots, A_{|\mathcal{E}|}]^T \in \mathbb{R}^{n|\mathcal{E}| \times n}$. Moreover, $\sum_{i \in \mathcal{E}} (A_i y)(c_i) = \text{sum}_{\text{column}}(A_y \mathbf{Diag}(c))$, where $\text{sum}_{\text{column}}(\bullet)$ returns a column vector containing the sum of each row. Similar operations can be implemented in the update of y. These structured matrices avoid loop computation when finding elements of H_x and b_x and significantly reduce the computational time. All computational techniques in $x-$update applies to the $y-$update in (3.27b).

For the s-update in (3.27c), as it is a constrained optimization problem, it is not as straightforward as x- and $y-$update. By reformulating (3.27c), it is equivalent to

$$s^{r+1} = \arg\min_{s \geq 0} \left\| s + x^T B_{\mathcal{I}} y - d + \frac{\lambda}{\rho_2} \right\|^2. \tag{3.29}$$

Problem in (3.29) is actually a projection of s onto the nonnegative space. Namely

$$s^{r+1} = \text{Proj}_{s \geq 0} - (x^T B_{\mathcal{I}} y - d + \frac{\lambda}{\rho_2}) \tag{3.30}$$

```
Algorithm 1: Customized ADMM for Problem (3.25).
Inputs: Problem parameters $A_0, A_{\mathcal{E}}, B_{\mathcal{I}}, c, d,$
$\alpha_1 \geq 1, \alpha_2 \geq 1, \alpha_3 \geq 1.$
Outputs: Local minimizer $(x, y, s)$ to (3.25)
Initialize: $(y^0, s^0, D^0)$ as random vectors and initial penalty coefficient $\rho_1^0, \rho_2^0, \rho_3^0$
for $r = 0, 1 \ldots$ do
    Update of $x$
        $H_x = \rho_3 I_n + \rho_1 \sum_{i\in\mathcal{E}} (A_i y^r)(A_i y^r)^T + \rho_2 \sum_{j\in\mathcal{I}} (B_j y^r)(B_j y^r)^T$
        $b_x = A_0 y^r + \nu^r - \rho_3 y^r + \sum_{i\in\mathcal{E}} (A_i y^r)(\mu_i - \rho_1 c_i) + \sum_{j\in\mathcal{I}} (B_j y^r)\left(\lambda_j + \rho_2 (s_j^r - d_j)\right)$
        $x^{r+1} = -H_x \setminus b_x.$    (3.32)
    Update $y$
        $H_y = \rho_3 I_n + \rho_1 \sum_{i\in\mathcal{E}} (A_i x^{r+1})(A_i x^{r+1})^T + \rho_2 \sum_{j\in\mathcal{I}} (B_j x^{r+1})(B_j x^{r+1})^T$
        $b_y = A_0 x^{r+1} - \nu^r - \rho_3 x^{r+1} + \sum_{i\in\mathcal{E}} (A_i x^{r+1})(\mu_i - \rho_1 c_i) + \sum_{j\in\mathcal{I}} (B_j x^{r+1})\left(\lambda_j + \rho_2 (s_j^r - d_j)\right)$
        $y^{r+1} = -H_y \setminus b_y.$    (3.33)
    Update $s$
        $s^{r+1} = \max\left(-\left((x^{r+1})^T B_{\mathcal{I}} y^{r+1} - d + \frac{\lambda^r}{\rho_2}\right), 0\right).$    (3.34)
    Update $(\mu, \lambda, \nu)$
        $\mu^{r+1} = \mu^r + \rho_1^r((x^{r+1})^T A_{\mathcal{E}} y^{r+1} - c).$
        $\lambda^{r+1} = \lambda^r + \rho_2^r((x^{r+1})^T B_{\mathcal{I}} y^{r+1} + s^{r+1} - d).$
        $\nu^{r+1} = \nu^r + \rho_3^r(x^{r+1} - y^{r+1}).$    (3.35)
    Update $\rho_1, \rho_2, \rho_3$
        $\rho_1^{r+1} = \alpha_1 \rho_1^r$
        $\rho_2^{r+1} = \alpha_2 \rho_2^r$
        $\rho_3^{r+1} = \alpha_3 \rho_3^r.$    (3.36)
    Calculate
        $\Delta = \max\left(\frac{\|x^{r+1} - y^{r+1}\|}{\max(1, \|x^{r+1}\|)}, \frac{\|(x^{r+1})^T A_{\mathcal{E}} y^{r+1} - c\|}{\max(1, \|c\|)}, \frac{\|(x^{r+1})^T B_{\mathcal{I}} y^{r+1} + s^{r+1} - d\|}{\max(1, \|d\|)}\right)$
    if $\Delta \leq \epsilon$ then , break
    end if
end for
Find $(x, y, s)$
```

which can be solved in the following closed form

$$s^{r+1} = \max\left(-\left(x^T B_{\mathcal{I}} y - d + \frac{\lambda}{\rho_2}\right), 0\right) \tag{3.31}$$

where the superscripts in x, y, and λ are omitted. The closed-form solution in s-update justifies the advantage of introducing the slack variable s in (3.24). By this point, we find closed-form solutions for all three subproblems and the customized ADMM is summarized in Algorithm 1. Note that unlike the standard ADMM using fixed penalty parameters, the increasing penalty parameters (ρ_1, ρ_2, ρ_3) are employed in Algorithm 1. Although a large penalty parameter generally contributes to faster convergence for the iterative algorithm, it makes the algorithm sensitive to the initial guesses and may lead to unsatisfactory performance. To balance the performance of the objective function and the convergence, the penalty parameters are set to increase gradually starting at small positive values.

IV. Convergence Analysis

In this section, we aim to prove the global convergence of Algorithms 1 under certain conditions. In what follows, similar to $D^r = (\mu^r, \lambda^r, \nu^r)$, we denote $P^r = (x^r, y^r, s^r)$ and $\rho^r = (\rho_1^r, \rho_2^r, \rho_3^r)$ as a collection of the primal variables and the penalty coefficients, respectively. First, the following assumption is introduced.

Assumption IV.1: The sequences $\{\mu^r, \lambda^r, \nu^r\}$ are bounded in norm.

Then, we have the lemma as follows.

Lemma IV.2: $\mathcal{L}(P^r; D^r; \rho^r)$ is bounded.

Proof: Recall the definition of $\mathcal{L}(P^k; D^k; \rho^k)$

$$\begin{aligned}\mathcal{L}(P^r, D^r, \rho^r) &= (x^r)^T A_0 y^r + \frac{\rho_1^r}{2}\left\|(x^r)^T A_{\mathcal{E}} y^r - c + \frac{\mu^r}{\rho_1^r}\right\|_F^2 \\ &+ \frac{\rho_2^r}{2}\left\|(x^r)^T B_{\mathcal{I}} y^r + s^r - d + \frac{\lambda^r}{\rho_2^r}\right\|_F^2 + \frac{\rho_3^r}{2}\left\|x^r - y^r + \frac{\nu^r}{\rho_3^r}\right\|_F^2 \\ &- \frac{1}{2\rho_1^r}\|\mu^r\|_F^2 - \frac{1}{2\rho_2^r}\|\lambda^r\|_F^2 - \frac{1}{2\rho_3^r}\|\nu^r\|_F^2 > -\infty,\end{aligned}$$

where the inequality follows the boundedness of $\{(x^r)^T A_0 y^r\}$ and $\{\mu^r, \lambda^r, \nu^r\}$ in Assumption IV.1. ■

Remark 1: Assumption IV.1 is less restrictive compared to the assumptions in existing works of ADMM for nonconvex optimization, where the convergence of the dual variables [53] and/or the primal variables [50], [52] are directly assumed. Furthermore, the boundedness of $\{(x^r)^T A_0 y^r\}$ can be achieved by adjusting the algorithmic parameters ρ_1 and ρ_2. In Algorithm 1, it states that $\{x^r\}$ and $\{y^r\}$ are obtained by solving a strongly convex quadratic programming in (3.32) and (3.33), respectively. Then, (3.32) shows that

$x^{r+1} = -H_x(A_i, B_j; \rho_1, \rho_2)\backslash b_x(A_i, B_j; \mu^r, \nu^r)$. As ρ_1 and ρ_2 are tunable parameters, x^{r+1} can be guaranteed to be bounded if ρ_1 and ρ_2 are appropriately selected, e.g., exponentials with higher order than μ^r and ν^r. Consequently, x^r (and similarly y^r) will be bounded and so will $\{(x^r)^T A_0 y^r\}$.

As it is already claimed that each subproblem is strongly convex with regard to x, y and s individually, the next lemma will quantify the strong convexity.

Lemma IV.3:

$$\nabla^2 \mathcal{L}_x^r \succeq \rho_3^r I,\ \nabla^2 \mathcal{L}_y^r \succeq \rho_3^r I,\ \nabla^2 \mathcal{L}_s^r \succeq \rho_2^r I.$$

Proof: From the definition of $\mathcal{L}(P^r; D^r; \rho^r)$

$$\begin{aligned}\frac{\partial^2 \mathcal{L}^r}{\partial x^2} &= \rho_1^r \sum_{i\in\mathcal{E}} A_i y^{r-1} \left(A_i y^{r-1}\right)^T \\ &\quad + \rho_2^r \sum_{j\in\mathcal{I}} B_j y^{r-1} (B_j y^{r-1})^T + \rho_3^r I \succeq \rho_3^r I. \quad (4.37)\end{aligned}$$

Similar conclusions can be obtained for $\nabla^2 \mathcal{L}_y^r$ and $\nabla^2 \mathcal{L}_s^r$. ■

With these introductory theorems, we then focus on proving convergence of Algorithm 1.

Theorem IV.4: If Assumption IV.1 holds and given a sequence $\{\rho_i^r\}_{r=1}^\infty, i = 1, 2, 3$ such that

$$\rho_i^r > 0,\ \sum \frac{\rho_i^{r+1}}{(\rho_i^r)^2} < \infty,\ \text{and} \sum \frac{1}{\rho_i^r} < \infty$$

then $\{x^r, y^r, s^r\}$ generated by Algorithm 1 will globally converge to a stationary point of (3.25).

Proof: For the update of x in (3.32) with $\{y^r, s^r, D^r, \rho^r\}$, we have

$$\begin{aligned}&\mathcal{L}(x^{r+1}, y^r, s^r; D^r; \rho^r) - \mathcal{L}(P^r; D^r; \rho^r) \\ &\quad \le \langle \nabla_x \mathcal{L}(x^{r+1}, y^r, s^r; D^r; \rho^r), x^{r+1} - x^r \rangle \\ &\quad\quad - \frac{\lambda_{\min}(\nabla_x^2 \mathcal{L}_r)}{2} \|x^{r+1} - x^r\|_F^2 \\ &\quad \le -\frac{\rho_3^r}{2} \|x^{r+1} - x^r\|_F^2 \quad (4.38)\end{aligned}$$

where the inequality follows from the first-order optimality condition of x and the ρ_3^r-strong convexity of $\mathcal{L}$ with respect to x in Lemma IV.3.

In parallel, for the update of y and s in (3.33) and (3.34), respectively, we have

$$\begin{aligned}&\mathcal{L}(x^{r+1}, y^{r+1}, s^r; D^r; \rho^r) - \mathcal{L}(x^{r+1}, y^r, s^r; D^r; \rho^r) \\ &\quad \le -\frac{\rho_3^r}{2} \|y^{r+1} - y^r\|_F^2 \quad (4.39)\end{aligned}$$

and

$$\begin{aligned}&\mathcal{L}(P^{r+1}; D^r; \rho^r) - \mathcal{L}(x^{r+1}, y^{r+1}, s^r; D^r; \rho^r) \\ &\quad \le -\frac{\rho_2^r}{2} \|s^{r+1} - s^r\|_F^2. \quad (4.40)\end{aligned}$$

In addition to the update of the dual variables in (3.35) and the penalty coefficients in (3.36), we have

$$\begin{aligned}&\mathcal{L}(P^{r+1}; D^{r+1}; \rho^{r+1}) - \mathcal{L}(P^{r+1}; D^r; \rho^r) \\ &= \langle \mu^{r+1} - \mu^r, (x^{r+1})^T A_{\mathcal{E}} y^{r+1} - c \rangle \\ &\quad + \langle \lambda^{r+1} - \lambda^r, (x^{r+1})^T B_{\mathcal{I}} y^{r+1} + s^{r+1} - d \rangle \\ &\quad + \langle \nu^{r+1} - \nu^r, x^{r+1} - y^{r+1} \rangle \\ &\quad + \frac{\rho_1^{r+1} - \rho_1^r}{2} (\|(x^{r+1})^T A_{\mathcal{E}} y^{r+1} - c\|_F^2) \\ &\quad + \frac{\rho_2^{r+1} - \rho_2^r}{2} (\|(x^{r+1})^T B_{\mathcal{I}} y^{r+1} + s^{r+1} - d\|_F^2) \\ &\quad + \frac{\rho_3^{r+1} - \rho_3^r}{2} (\|x^{r+1} - y^{r+1}\|_F^2) \\ &= \frac{\rho_1^{r+1} + \rho_1^r}{2(\rho_1^r)^2} \|\mu^{r+1} - \mu^r\|_F^2 + \frac{\rho_2^{r+1} + \rho_2^r}{2(\rho_2^r)^2} \|\lambda^{r+1} - \lambda^r\|_F^2 \\ &\quad + \frac{\rho_3^{r+1} + \rho_3^r}{2(\rho_3^r)^2} \|\nu^{r+1} - \nu^r\|_F^2 \quad (4.41)\end{aligned}$$

where the first equality follows the definition of $\mathcal{L}$ and the second follows (3.35). Moreover, it can be verified that

$$\begin{aligned}&\mathcal{L}(P^{r+1}; D^{r+1}; \rho^{r+1}) - \mathcal{L}(P^r; D^r; \rho^r) \\ &= \mathcal{L}(x^{r+1}, y^r, s^r; D^r; \rho^r) - \mathcal{L}(P^r; D^r; \rho^r) \\ &\quad + \mathcal{L}(x^{r+1}, y^{r+1}, s^r; D^r; \rho^r) - \mathcal{L}(x^{r+1}, y^r, s^r; D^r; \rho^r) \\ &\quad + \mathcal{L}(P^{r+1}; D^r; \rho^r) - \mathcal{L}(x^{r+1}, y^{r+1}, s^r; D^r; \rho^r) \\ &\quad + \mathcal{L}(P^{r+1}; D^{r+1}; \rho^{r+1}) - \mathcal{L}(P^{r+1}; D^r; \rho^r). \quad (4.42)\end{aligned}$$

By incorporating (4.38)–(4.41), (4.42) can be rewritten as

$$\begin{aligned}&\mathcal{L}(P^{r+1}; D^{r+1}; \rho^{r+1}) - \mathcal{L}(P^r; D^r; \rho^r) \\ &\quad \le -\kappa_1^r \|x^{r+1} - x^r\|_F^2 - \kappa_2^r \|y^{r+1} - y^r\|_F^2 \\ &\quad\quad - \kappa_3^r \|s^{r+1} - s^r\|_F^2 + \eta_1^r \|\mu^{r+1} - \mu^r\|_F^2 \\ &\quad\quad + \eta_2^r \|\lambda^{r+1} - \lambda^r\|_F^2 + \eta_3^r \|\nu^{r+1} - \nu^r\|_F^2\end{aligned}$$

where $\eta_i^r = \frac{\rho_i^{r+1} + \rho_i^r}{2(\rho_i^r)^2}$, $i = 1, 2, 3$, $\kappa_1^r = \kappa_2^r = \frac{\rho_3^r}{2} > 0$, and $\kappa_3^r = \frac{\rho_2^r}{2} > 0$. By telescoping $\sum_{r=0}^{R-1} (\mathcal{L}(P^{r+1}; D^{r+1}; \rho^{r+1}) - \mathcal{L}(P^r; D^r; \rho^r))$, we have

$$\begin{aligned}&\sum_{r=0}^{R-1} \mathcal{L}(P^{r+1}; D^{r+1}; \rho^{r+1}) - \mathcal{L}(P^r; D^r; \rho^r) \\ &\quad = \mathcal{L}(P^R; D^R; \rho^R) - \mathcal{L}(P^0; D^0; \rho^0) \\ &\quad \le \sum_{r=0}^{R-1} \eta^r \Bigg(\|\mu^{r+1} - \mu^r\|_F^2 + \|\lambda^{r+1} - \lambda^r\|_F^2 \\ &\quad\quad + \|\nu^{r+1} - \nu^r\|_F^2 \Bigg) \\ &\quad\quad - \sum_{r=0}^{R-1} \kappa^r \Bigg(\|x^{r+1} - x^r\|_F^2 - \|y^{r+1} - y^r\|_F^2 \\ &\quad\quad - \|s^{r+1} - s^r\|_F^2 \Bigg) \quad (4.43)\end{aligned}$$

with $\kappa^r = \min(\kappa_1^r, \kappa_2^r, \kappa_3^r) > 0$ and $\eta^r = \max(\eta_2^r, \eta_2^r, \eta_3^r) > 0$. For the first term in (4.43), it can be verified that

$$0 \leq \sum_{r=0}^{R-1} \eta^r \Bigg(\|\mu^{r+1} - \mu^r\|_F^2 + \|\lambda^{r+1} - \lambda^r\|_F^2 + \|\nu^{r+1} - \nu^r\|_F^2 \Bigg) \leq 12 \sum_{r=0}^{R-1} \eta^r B \tag{4.44}$$

where $\{\mu^r, \lambda^r, \nu^r\}$ is bounded from the statement of Assumption IV.1. In other words, there exists $B < +\infty$ such that $B \geq \max\{\|\mu^r\|_F^2, \|\lambda^r\|_F^2, \|\nu^r\|_F^2\}, \ \forall\, r = 1, \ldots, R-1$. Additionally, recall that $\eta_i^r = \frac{\rho_i^{r+1} + \rho_i^r}{2(\rho_i^r)^2}, i = 1, 2, 3, \ \ \eta^r = \max(\eta_1^r, \eta_2^r, \eta_3^r) > 0$, and $\sum \frac{\rho_i^{r+1} + \rho^r}{2(\rho_i^r)^2} \leq \sum \frac{\rho_i^{r+1}}{(\rho_i^r)^2} < +\infty$, then it leads to $\sum \eta_i^r < +\infty$. Then, $\sum_{r=0}^{R-1} \eta^r B$ is bounded, which implies that

$$0 \leq \quad \sum_{r=0}^{R-1} \eta^r \Bigg(\|\mu^{r+1} - \mu^r\|_F^2 + \|\lambda^{r+1} - \lambda^r\|_F^2 + \|\nu^{r+1} - \nu^r\|_F^2 \Bigg) \leq +\infty. \tag{4.45}$$

Since $\mathcal{L}(P^r; D^r; \rho^r)$ is bounded, it follows that the second term in (4.43) is bounded by

$$0 \leq \sum_{k=0}^{R-1} \kappa^r \Bigg(\|x^{r+1} - x^r\|_F^2 + \|y^{r+1} - y^r\|_F^2 + \|s^{r+1} - s^r\|_F^2 \Bigg) \leq +\infty. \tag{4.46}$$

As $\kappa^r > 0$, then combing the above analysis yields $x^{r+1} - x^r \to 0, y^{r+1} - y^r \to 0, s^{r+1} - s^r \to 0$. Moreover

$$\sum_{k=1}^{\infty} \frac{1}{\rho_i^r} < \infty \iff \frac{1}{\rho_i^r} \to 0 \iff \rho_i^r \to \infty, i = 1, 2, 3.$$

Again from Assumption IV.1, i.e., the boundedness of the dual variables, it implies that

$$(x^{r+1})^T A_{\mathcal{E}} y^{r+1} - c = \frac{1}{\rho_1^r}(\mu^{r+1} - \mu^r) \to 0$$

$$(x^{r+1})^T B_{\mathcal{I}} y^{r+1} + s^{r+1} - d = \frac{1}{\rho_2^r}(\lambda^{r+1} - \lambda^r) \to 0$$

$$x^{r+1} - y^{r+1} = \frac{1}{\rho_3^r}(\nu^{r+1} - \nu^r) \to 0.$$

Therefore, the limit points x^*, y^*, and s^* are all feasible. By simply checking the first optimality condition, it verifies that this accumulation point is a stationary point of (3.25). ■

Remark 2: Assumption IV.1 is used in Lemma IV.2 and (4.44). Actually, the requirement for the latter is milder as the difference between adjacent sequence, i.e., $\{\mu^{r+1} - \mu^r\}$, is required to be bounded instead of $\{\mu^r\}$; similarly for ν and λ. As a result, in the worst case, if Assumption IV.1, i.e., the boundedness on sequences $\{\mu^r, \lambda^r, \nu^r\}$, cannot be satisfied, we can at least prove the local convergence as long as that $\{\mu^{r+1} - \mu^r, \lambda^{r+1} - \lambda^r, \nu^{r+1} - \nu^r\}$ are bounded. This bounded difference of adjacent sequence can be obtained via our updating rules. Namely, starting with a point, where the primal infeasibilities, i.e., $(x^{r+1})^T A_{\mathcal{E}} y^{r+1} - c, (x^{r+1})^T B_{\mathcal{I}} y^{r+1} + s^{r+1} - d, x^{r+1} - y^{r+1}$, are sufficiently small, the above bounded difference of adjacent sequence can be satisfied based on the update rule of the dual variables in (3.35) with fine-tuned parameters $(\rho_i, i = 1, 2, 3)$ and thus local convergence is guaranteed.

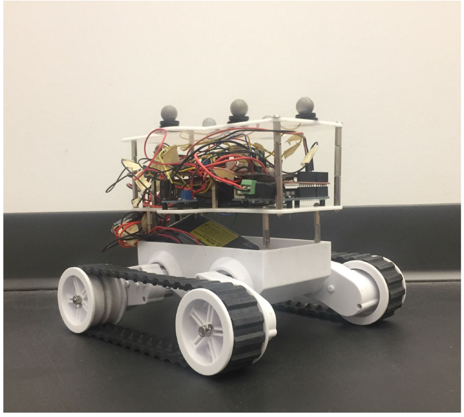


Fig. 3. Demonstration UGV.

V. Simulation and Experimental Results

In this section, we apply the proposed algorithm in the ITMP of TSP and hybrid control of the rendezvous and docking problem under different scenarios. The settings of numerical simulation as well as the experimental testbed are described first. Then comparative numerical results to the state-of-art optimization solver are presented to verify the effectiveness, efficacy, and robustness of the proposed algorithm. Moreover, on-board vehicle motions generated from the implemented algorithm are presented to further validate its real-time computational capability in solving decision-making problems at this scale. All numerical simulation is run in MATLAB environment on a standard Dell desktop with a Intel Xeon CPU that has a 3.6 GHz processor and a 32.0 GB memory.

A. Equipment, Environment, and Vehicle Design

The simulation and experimental cases use both the performance characteristics and experimental data gathered by a demonstration UGV, shown in Fig. 3. The demonstration UGV is based on a Dagu 5 robot chassis [56] controlled by an Arduino Uno [57] and a wireless modem. This vehicle makes use of two motors that can be controlled independently, enabling the vehicle to traverse a continuous path.

Based on the characteristics of the demonstration UGV, the unicycle model presented in (2.3) for the specific vehicle in Fig. 4 is written as

$$\begin{aligned} \dot{x} &= V_{\text{const}} \cos(\theta) \\ \dot{y} &= V_{\text{const}} \sin(\theta) \\ \dot{\theta} &= \frac{V_r + V_l}{L} \end{aligned}$$

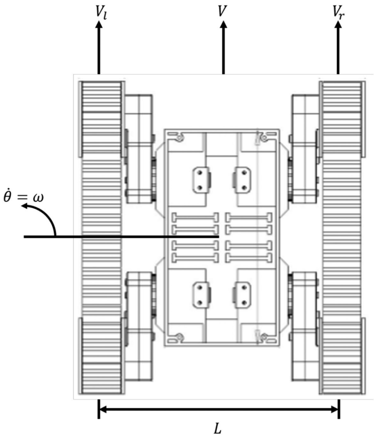


Fig. 4. Linear and angular speed of the demonstration UGV.

where L represents the distance between the vehicle tracks/wheels and $V_{\text{const}} = \frac{V_r + V_l}{2}$ with V_l and V_r representing the left and right wheel speed, respectively. Speed of both wheels is controlled by adjusting the pulsewidth modulation (PWM) of the voltage supplied to the corresponding motor. The PWM number is an integer which ranges from 0 to 250 and each PWM number corresponds to a fixed wheel speed. Based on this model, a linear velocity V_{const} and angular velocity $\dot{\theta}$ from the optimized results can be related to the V_r and V_l of the vehicle, allowing it to follow the prescribed trajectory. It was determined experimentally that $V_{\text{const}} = 0.219$ m/s and $\dot{\theta}_{\max} = 0.31$ rad/s for the demonstration vehicle. Finally, all experimental tests were conducted in an indoor environment of a 3.5×3.5 m^2 area that uses a VICON [58] motion capture system to provide information about the vehicle's location and orientation.

B. Virtual Simulation Examples of ITMP

We first consider a simple case for the ITMP of TSP. The traditional hierarchical framework employs the exhaustive search in the task planning level to determine the visiting sequence, which is followed by the motion planning to generate optimal paths between any two connected waypoints. However, this hierarchical framework only applies to small-scale problems due to its limited scalability. Considering P waypoints (excluding the starting point), there are $P!$ combinations for the traveling sequence. Also, even under a specified traveling sequence, it still requires solving P nonconvex motion planning problems to determine the traveling paths. As the final reformulated problem in (2.13) is a mixed-integer nonlinear programming problem, existing solvers for mixed-integer linear programming, such as CPLEX [59], are not applicable. For mixed-integer nonlinear solvers, such as OQNLP, minlpBB, and minlpSolve, embedded in commercial optimization software [60], the complex constraints in this challenging problem make them very sensitive to the initial guess and difficult to converge under random initials. Finally, we apply the state-of-art mixed-integer nonconvex programming solver Gurobi [61] and nonlinear solver [62] SNOPT to solve the same QCQP problem and compare its results with those generated from Algorithm 1 in terms of objective value, efficacy and robustness.

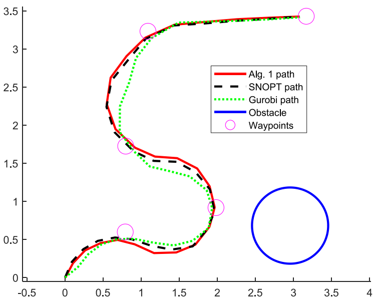


Fig. 5. Comparison of paths generated from Algorithm 1, SNOPT and Gurobi.

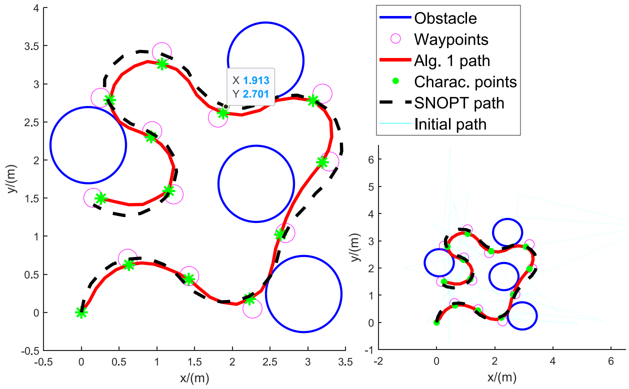


Fig. 6. Comparison of paths generated from Algorithm 1 and SNOPT.

TABLE I
COMPARISON BETWEEN THE ALGORITHM 1, SNOPT, AND GUROBI FOR ITMP SCENARIO ONE (5 WAYPOINTS, 1 OBSTACLE)

	computation time / sec	objective value	convergence rate
SNOPT	16.92	36.50	32/100
Algorithm 1	0.89	35.64	98/100
Gurobi	3882.95	33.81	24/100

To evaluate the robustness, we run both methods for 100 trials with random initials. The initial guess for binary decision variables is randomly generated within the interval [0, 1]. To guarantee random distribution of initials, we have that $|\Gamma_{\text{initial}} - \text{round}(\Gamma_{\text{initial}})|/|\text{round}(\Gamma_{\text{initial}})| = 0.299$, which is the mean value of 100 random initials. For the initial guess of continuous variables, one of the 100 trails is illustrated in the right plot of Fig. 6. The number of discrete segments between two adjacent waypoints is $m = 6$ and we set two scenarios of different scales: one has $P = 5$ waypoints and $n_A = 1$ obstacle, while the other has $P = 12$ waypoints and $n_A = 4$ obstacles. Trajectories of representative convergence cases for two scenarios are presented in Figs. 5 and 6, respectively. In addition, the comparative results of multiple trails for the two scenarios are demonstrated in Tables I and II, respectively, and note that the computation time as well as the objective value are the mean value of all converged cases. It indicates that the proposed algorithm outperforms SNOPT with significantly improved performance in all of the three categories. While Gurobi can solve the small scale problem with a slightly reduced objective value, the computational time is much longer than that required by the

TABLE II
COMPARISON BETWEEN THE ALGORITHM 1, SNOPT, AND GUROBI FOR ITMP SCENARIO ONE (12 WAYPOINTS, 4 OBSTACLE). "NA" DENOTES "NOT APPLICABLE"

	computation time / sec	objective value	convergence rate
SNOPT	107.07	62.90	3/100
Algorithm 1	5.03	51.68	95/100
Gurobi	$> 36,000$	NA	NA

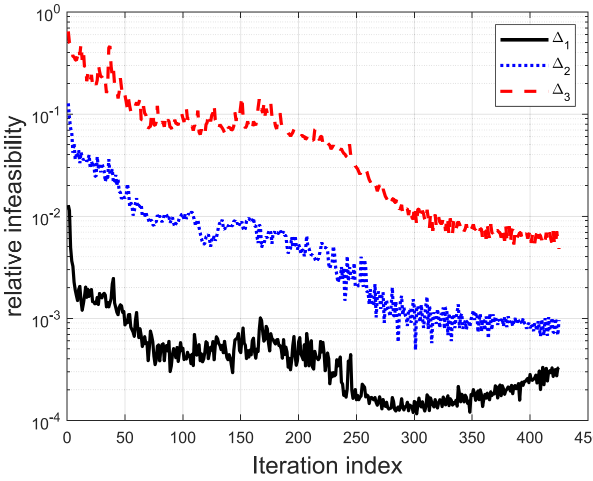


Fig. 7. Convergence history of the relative infeasibility of the three coupling constraints in (3.25), where $\Delta_1 = \frac{\|(x^{r+1})^T A_{\mathcal{E}} y^{r+1} - c\|}{\max(1,\|c\|)}$, $\Delta_2 = \frac{\|(x^{r+1})^T B_{\mathcal{I}} y^{r+1} + s^{r+1} - d\|}{\max(1,\|d\|)}$, $\Delta_3 = \frac{\|\|x^{r+1} - y^{r+1}\|\|}{\max(1,\|x^{r+1}\|)}$.

proposed algorithm. Furthermore, Gurobi is not applicable to large scale mixed-integer nonconvex QCQPs. It cannot converge or even find a feasible solution for the case $P = 12$ and $n_A = 4$ under a given time limit, i.e., 10 h. As Gurobi solves the mixed-integer nonconvex QCQP by the Branch and Bound method that has exponential computation complexity, the scalability of Gurobi is not guaranteed. The comparison with alternative solvers further shows the advantage of our proposed algorithm in solving large-scale real-time mixed-type decision making problems. Moreover, the convergence history of the relative infeasibility of the three coupling constraints generalized in (3.25) are presented in Fig. 7 for one case selected from the 100 trials.

C. Experimental Examples of ITMP

Next, we aim to verify the computational capability of the proposed algorithm in generating on-board decisions to maneuver the vehicle in real time. Two scenarios for ITMP of the TSP are considered here. One scenario will assign a new mission, e.g., a new set of waypoints, after finishing the current mission or during the process of a mission. The other scenario considers new discovered or moving obstacles. In both scenarios, decisions are required to be regenerated on-board once the mission definitions or constraints are updated. For scenario one, a human operator assigns the new waypoints as inputs into the algorithm to generate new decisions. Both assigned missions have $P = 5$ and $m = 6$. For the second scenario, tracking markers are attached on the surface of all obstacle objects such that their motions can be tracked by the Vicon tracking system in real time. Their locations are updated within a specific time interval and new decisions will be generated once locations of obstacles are updated. These decisions will be converted into vehicle maneuvering commands and sent to the demonstration UGV via the wireless modem.

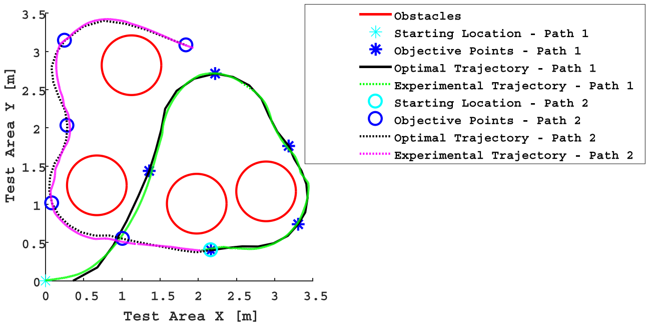


Fig. 8. Simulated and experimental paths for scenario one.

TABLE III
COMPARISON BETWEEN ALGORITHM 1 AND SNOPT FOR SCENARIOS ONE AND TWO WHERE EACH SCENARIO HAS TWO MISSIONS AND THUS THE CORRESPONDING TIME AND OBJECTIVE VALUE HAVE TWO SEGMENTS

Scenario one	computation time/sec	objective value	$\Delta(\%)$
SNOPT	39.13 / 12.97	38.59 / 29.89	N/A
Alg. 1	0.47 / 0.43	34.43 / 27.72	4.21
Scenario two	computation time/sec	objective value	$\Delta(\%)$
SNOPT	17.36 / 5.85	40.15 / 22.40	N/A
Alg. 1	0.56 / 0.25	38.72 / 21.28	2.45

For scenario one, the example presented here requires the UGV to travel to a new set of waypoints once the current mission is finished. The generated paths for two assigned missions from Algorithm 1 and the experimental paths of the UGV conducting the real-time received commands are shown in Fig. 8. Moreover, the comparison to results from SNOPT in terms of computation time and objective value is shown in Table III, where $\Delta(\%)$ represents the relative error between the path planned by Algorithm 1 and the corresponding one executed by the UGV. In both performance categories, Algorithm 1 displays its obvious advantages. Within less than one second, Algorithm 1 generates converged solution for each waypoints-visiting mission. The low $\Delta(\%)$ value for both scenarios indicates accuracy of the problem model. For the results generated from SNOPT, we choose the one with best convergence property among multiple initial inputs. Paths generated from the proposed algorithm for scenario two are shown in Fig. 9. Similarly, the comparative results in Table III are provided to verify that the proposed algorithm outperforms SNOPT. Again, the result presented for SNOPT is the one with best convergence property among multiple initial inputs. For the second scenario in Table III, new decisions can also be regenerated within less than one second from the proposed algorithm in Matlab computation environments. The computational time

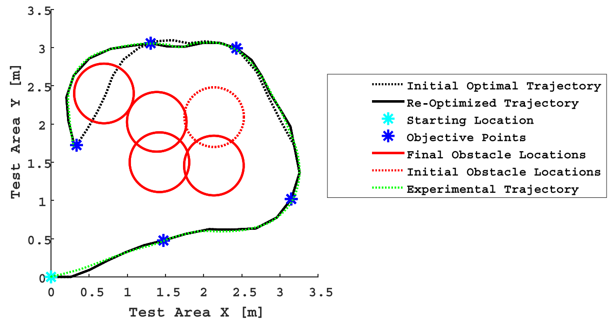


Fig. 9. Simulated and experimental paths for scenario two.

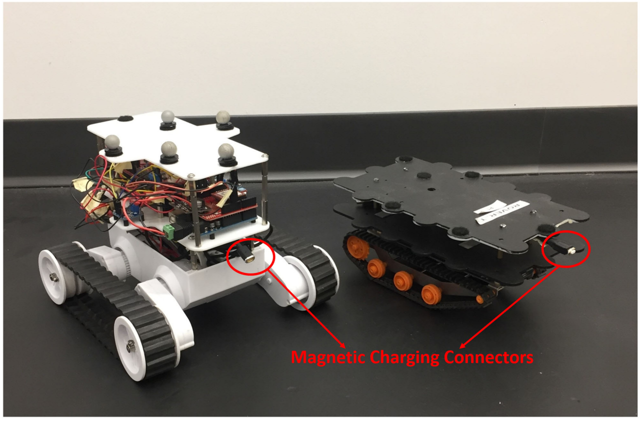


Fig. 10. Prototype of the demonstration vehicle and the charging station.

can be further reduced when implementing the algorithm in C or C++. In other words, the algorithm demonstrates high implementability for solving real-time mixed-type decision-making problems at a similar scale.

D. Virtual and Experimental Example of the Rendezvous and Docking Problem

For the rendezvous and docking problem, a charging station is built besides the demonstration UGV described above. Fig. 10 shows the prototypes of the demonstration UGV and the charging station. Upon proximity with the field-of-view cone set as $\psi = \pi/3$, they can be docked together via the magnetic charging connectors. We assume the UGV operates at a relatively slow speed and it can instantaneously switch from one speed to another by adjusting PWM numbers of both wheels. Therefore, the dynamic effects during the speed switches are ignored. As our focus is to demonstrate the computational performance of the proposed algorithm, the electromechanical design and sensing components, such as the range sensor, of both vehicles are omitted here as our purpose is to verify the computational capability of the proposed algorithm. The generated path and the experimental results are presented in Fig. 11. Additionally, Table IV gives the comparative results to validate the advantages of Algorithm 1 against SNOPT. Again, the proposed algorithm is capable of generating the optimal hybrid control commands within one second with much reduced objective value. Furthermore, the experimental results verify the accuracy of the modeling scheme in form of QCQP. Videos for all experimental cases are provided.

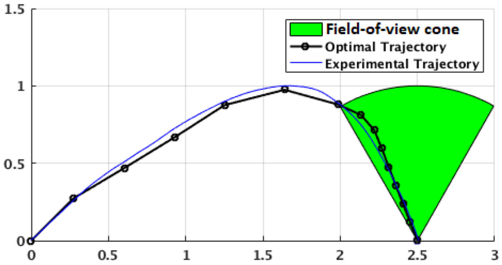


Fig. 11. Simulated and experimental paths for the rendezvous and docking problem.

TABLE IV
COMPARISON BETWEEN ALGORITHM 1 AND SNOPT FOR THE RENDEZVOUS AND DOCKING PROBLEM

	computation time/sec	objective value	$\Delta(\%)$
SNOPT	18.14	31.12	N/A
Alg. 1	0.48	23.44	1.74

VI. CONCLUSION

This article investigated a class of mixed-type decision-making problems in robotic systems. The proposed method can be applied to solve a wide range of ITMP problems and hybrid control problems, where both discrete and continuous decision variables are included. These problems are formulated as general/nonconvex QCQPs through discretization and equivalent conversions. An efficient optimization algorithm based on ADMMs is proposed to solve QCQPs. Moreover, convergence proof of the proposed algorithm under certain conditions is provided. Extensive comparison to the state-of-art optimization solver verifies the effectiveness, efficacy as well as robustness of the proposed algorithm. We further demonstrated the computational capability by implementing the algorithm in a closed-loop to generate real-time commands upon new assigned missions and updated constraints. The generated commands are sent to an experimental vehicle to verify the modeling and computational accuracy. Future research directions will extend this work to a multirobot team for cooperative control by developing distributed optimization algorithms. Furthermore, more challenging decision-making problems considering complex manipulations will be investigated.

REFERENCES

[1] J. Hoffmann and B. Nebel, “The FF planning system: Fast plan generation through heuristic search,” *J. Artif. Intell. Res.*, vol. 14, pp. 253–302, 2001.

[2] H. Kautz and B. Selman, “Unifying sat-based and graph-based planning,” in *Proc. Int. Joint Conf. Artif. Intell.*, 1999, vol. 99, pp. 318–325.

[3] M. Tenorth and M. Beetz, “Representations for robot knowledge in the knowrob framework,” *Artif. Intell.*, vol. 247, pp. 151–169, 2017.

[4] L. Blackmore, M. Ono, and B. C. Williams, “Chance-constrained optimal path planning with obstacles,” *IEEE Trans. Robot.*, vol. 27, no. 6, pp. 1080–1094, Dec. 2011.

[5] S. M. LaValle and J. J. Kuffner Jr, "Rapidly-exploring random trees: Progress and prospects," 2000.

[6] S. e. a. Karaman, "Anytime motion planning using the RRT*," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1478–1483.

[7] J. Schulman *et al.*, "Motion planning with sequential convex optimization and convex collision checking," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1251–1270, 2014.

[8] E. M. Wolff, U. Topcu, and R. M. Murray, "Optimization-based trajectory generation with linear temporal logic specifications," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 5319–5325.

[9] C. Dornhege, P. Eyerich, T. Keller, S. Trüg, M. Brenner, and B. Nebel, "Semantic attachments for domain-independent planning systems," in *Proc. Towards Serv. Robots Everyday Environ.*, 2012, pp. 99–115.

[10] J. A. Wolfe, B. Marthi, and S. J. Russell, "Combined task and motion planning for mobile manipulation," in *Proc. Int. Conf. Automated Plan. Scheduling*, 2010, pp. 254–258.

[11] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, "Combined task and motion planning through an extensible planner-independent interface layer," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 639–646.

[12] L. P. Kaelbling and T. Lozano-Pérez, "Integrated task and motion planning in belief space," *Int. J. Robot. Res.*, vol. 32, no. 9/10, pp. 1194–1227, 2013.

[13] L. De Silva, M. Gharbi, A. K. Pandey, and R. Alami, "A new approach to combined symbolic-geometric backtracking in the context of human-robot interaction," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 3757–3763.

[14] M. Gharbi, R. Lallement, and R. Alami, "Combining symbolic and geometric planning to synthesize human-aware plans: Toward more efficient combined search," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 6360–6365.

[15] S. Moon, E. Oh, and D. H. Shim, "An integral framework of task assignment and path planning for multiple unmanned aerial vehicles in dynamic environments," *J. Intell. Robot. Syst.*, vol. 70, no. 1–4, pp. 303–313, 2013.

[16] T. Lozano-Pérez and L. P. Kaelbling, "A constraint-based method for solving sequential manipulation planning problems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 3684–3691.

[17] F. Lagriffoul, D. Dimitrov, A. Saffiotti, and L. Karlsson, "Constraint propagation on interval bounds for dealing with geometric backtracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 957–964.

[18] F. Lagriffoul, D. Dimitrov, J. Bidot, A. Saffiotti, and L. Karlsson, "Efficiently combining task and motion planning using geometric constraints," *Int. J. Robot. Res.*, vol. 33, no. 14, pp. 1726–1747, 2014.

[19] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "FFROB: An efficient heuristic for task and motion planning," in *Proc. Algorithmic Found. Robot. XI*, 2015, pp. 179–195.

[20] N. T. Dantam, Z. K. Kingston, S. Chaudhuri, and L. E. Kavraki, "Incremental task and motion planning: A constraint-based approach," in *Proc. Robot., Sci. Syst.*, 2016, pp. 1–6.

[21] C. Zhang and J. A. Shah, "Co-optimizing task and motion planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 4750–4756.

[22] M. Toussaint, "Logic-geometric programming: An optimization-based approach to combined task and motion planning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 1930–1936.

[23] S. Kiesel, E. Burns, C. M. Wilt, and W. Ruml, "Integrating vehicle routing and motion planning," in *Proc. 22nd Int. Conf. Automat. Plann. Schedul*, 2012.

[24] E. Edison and T. Shima, "Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms," *Comput. Oper. Res.*, vol. 38, no. 1, pp. 340–356, 2011.

[25] M. S. Cons, T. Shima, and C. Domshlak, "Integrating task and motion planning for unmanned aerial vehicles," *UnmannedSyst.*, vol. 2, no. 1, pp. 19–38, 2014.

[26] J. Lygeros, C. Tomlin, and S. Sastry, "Hybrid systems: Modeling, analysis and control," *Tech. Rep. UCB/ERL M99/34 Electron. Res. Lab., Univ. California Berkeley*, Berkeley, CA, USA, 1999.

[27] M. S. Branicky, V. S. Borkar, and S. K. Mitter, "A unified framework for hybrid control: Model and optimal control theory," *IEEE Trans. Autom. Control*, vol. 43, no. 1, pp. 31–45, Jan. 1998.

[28] H. Lin and P. J. Antsaklis, "Stability and stabilizability of switched linear systems: A survey of recent results," *IEEE Trans. Autom. Control*, vol. 54, no. 2, pp. 308–322, Feb. 2009.

[29] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE Trans. Robot.*, vol. 25, no. 6, pp. 1370–1381, Dec. 2009.

[30] W. G. Walter, *The Living Brain*. New York, NY, USA: Norton, 1953.

[31] M. C. Silverman, D. Nies, B. Jung, and G. S. Sukhatme, "Staying alive: A docking station for autonomous robot recharging," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, vol. 1, pp. 1050–1055.

[32] R. Cassinis, F. Tampalini, P. Bartolini, and R. Fedrigotti, "Docking and charging system for autonomous mobile robots," *Dept. Electron. Autom., Univ. Brescia, Brescia, Italy*, 2005.

[33] R. C. Luo, C. T. Liao, K. L. Su, and K. C. Lin, "Automatic docking and recharging system for autonomous security robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2005, pp. 2953–2958.

[34] Y.-C. Wu, M.-C. Teng, and Y.-J. Tsai, "Robot docking station for automatic battery exchanging and charging," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2009, pp. 1043–1046.

[35] A. Drenner and N. Papanikolopoulos, "Docking station relocation for maximizing longevity of distributed robotic teams," in *Proc. IEEE Int. Conf. Autom.*, 2006, pp. 2436–2441.

[36] N. Navarro-Guerrero, C. Weber, P. Schroeter, and S. Wermter, "Real-world reinforcement learning for autonomous humanoid robot docking," *Robot. Auton. Syst.*, vol. 60, no. 11, pp. 1400–1407, 2012.

[37] N. Mathew, S. L. Smith, and S. L. Waslander, "Multirobot rendezvous planning for recharging in persistent tasks," *IEEE Trans. Robot.*, vol. 31, no. 1, pp. 128–142, Feb. 2015.

[38] Z. Luo, W. Ma, A. M.-C. So, Y. Ye, and S. Zhang, "Nonconvex quadratic optimization, semidefinite relaxation, and applications," *IEEE Signal Process. Mag. (Special Issue on Convex Optimization for Signal Processing)*, 2010.

[39] X. Bao, N. Sahinidis, and M. Tawarmalani, "Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons," *Math. Program.*, vol. 129, pp. 129–157, 2011.

[40] B. G. A. K. O. Mehanna, K. Huang, and N. D. Sidiropoulos, "Feasible point pursuit and successive approximation of non-convex QCQPs," *IEEE Signal Process. Lett.*, vol. 22, no. 7, pp. 804–808, Jul. 2015.

[41] C. Sun and R. Dai, "An iterative approach to rank minimization problems," in *Proc. IEEE 54th Annu. Conf. Control Decis.*, 2015, pp. 3317–3323.

[42] S. Sojoudi and J. Lavaei, "Exactness of semidefinite relaxations for nonlinear optimization problems with underlying graph structure," *SIAM J. Optim.*, vol. 24, no. 4, pp. 1746–1778, 2014.

[43] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM J. Optim.*, vol. 11, no. 3, pp. 796–817, 2001.

[44] J. Eckstein and D. P. Bertsekas, "On the Douglas Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, no. 1, pp. 293–318, 1992.

[45] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[46] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM J. Optim.*, vol. 26, no. 1, pp. 337–364, 2016.

[47] G. Li and T. K. Pong, "Global convergence of splitting methods for nonconvex composite optimization," *SIAM J. Optim.*, vol. 25, no. 4, pp. 2434–2460, 2015.

[48] S. Lu, M. Hong, and Z. Wang, "A nonconvex splitting method for symmetric nonnegative matrix factorization: Convergence analysis and optimality," *IEEE Trans. Signal Process.*, vol. 65, no. 12, pp. 3120–3135, Jun. 2017.

[49] Y. Xu, W. Yin, Z. Wen, and Y. Zhang, "An alternating direction algorithm for matrix completion with nonnegative factors," *Front. Math. China*, vol. 7, no. 2, pp. 365–384, 2012.

[50] B. Jiang, S. Ma, and S. Zhang, "Alternating direction method of multipliers for real and complex polynomial optimization models," *Optimization*, vol. 63, no. 6, pp. 883–898, 2014.

[51] Y. Shen, Z. Wen, and Y. Zhang, "Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization," *Optim. Methods Softw.*, vol. 29, no. 2, pp. 239–263, 2014.

[52] K. Huang and N. D. Sidiropoulos, "Consensus-ADMM for general quadratically constrained quadratic programming," *IEEE Trans. Signal Process.*, vol. 64, no. 20, pp. 5297–5310, Oct. 2016.

[53] S. Magnússon, P. C. Weeraddana, M. G. Rabbat, and C. Fischione, "On the convergence of alternating direction lagrangian methods for nonconvex structured optimization problems," *IEEE Control Netw. Syst.*, vol. 3, no. 3, pp. 296–309, Sep. 2016.

[54] C. Sun, R. Dai, and P. Lu, "Multi-phase spacecraft mission optimization by quadratically constrained quadratic programming," in *Proc. AIAA Scitech 2019 Forum*, 2019, Art. no. 1667.

[55] S. Burer and H. Dong, "Representing quadratically constrained quadratic programs as generalized copositive programs," *Oper. Res. Lett.*, vol. 40, no. 3, pp. 203–206, 2012.

[56] "Dagu 5 robot chassis," 2018. [Online]. Available: http://www.pololu.com/product/1551

[57] "What Arduino can do," 2018. [Online]. Available: http://arduino.cc/

[58] "Affordable motion capture for any application," 2018. [Online]. Available: http://www.vicon.com/System/Bonita

[59] I. I. CPLEX, "V12. 1: User's manual for CPLEX," *Int. Bus. Mach. Corporation*, vol. 46, no. 53, 2009, Art. no. 157.

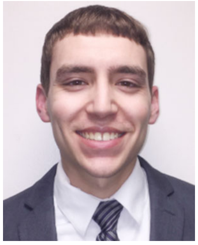
[60] K. Holmström, "The TOMLAB optimization environment in MATLAB," 1999.

[61] E. Rothberg, "Gurobi—the fastest solver—Gurobi," 2020.

[62] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Rev.*, vol. 47, no. 1, pp. 99–131, 2005.

Chuangchuang Sun received the B.S. degree from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2013, and the Ph.D. degree from the Ohio State University, Columbus, OH, USA, in 2018, both in aerospace engineering.

He is currently a Postdoctoral Associate with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA. His research interests focus on control, optimization, reinforcement learning with applications in robotics and space systems.



Nathaniel Kingry received the bachelor's degree in aerospace engineering from Iowa State University, Ames, IA, USA, in 2017, and the master's degree in aeronautics and astronautics from The Ohio State University, Columbus, OH, USA, in 2018.

He is currently a Flight Control Engineer with Rockwell Collins, Cedar Rapids, IA, USA.

Ran Dai received the B.S. degree in automation science from Beihang University, Beijing, China, in 2002, and the M.S. and Ph.D. degrees in aerospace engineering from Auburn University, Auburn, AL, USA, in 2005 and 2007, respectively.

She is currently an Associate Professor with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN, USA. Her research focuses on control of autonomous systems, numerical optimization, and networked dynamical systems.

Dr. Dai is an Associate Editor for the IEEE TRANSACTION ON AEROSPACE AND ELECTRONIC SYSTEMS, and a recipient of the National Science Foundation Career Award and NASA Early Faculty Career Award.