Motion Planning with Homotopy Class Constraints via the Auxiliary Energy Reduction Technique

Wenbo He, Yunshen Huang and Shen Zeng

Abstract— **We introduce the so-called Auxiliary Energy Reduction (AER) technique, which is a gradient-based approach to solving motion planning problems with homotopy class constraints for system models with full-scale nonholonomic dynamics. The hallmark of our approach is that we first introduce virtual control terms to the original system dynamics that ensure that any preset state trajectory is dynamically feasible with respect to the new extended system. We then gradually shift the contribution of the artificial inputs to the actual original inputs by solving a sequence of associated quadratic programs. When the contribution of the artificial inputs has been fully removed, the preset trajectory will have been deformed to a trajectory of the same homotopy class that is now also feasible with respect to the original system. The practicality of our method is demonstrated in simulation examples for the Brockett integrator, the unicycle, and a 12-dimensional nonlinear quadcopter model.**

I. Introduction

The basic task of motion planning is to find a dynamically feasible system trajectory connecting a given starting point with a desired target point in a state space that possibly contains one or more obstacles. Many control techniques can be leveraged to achieve this goal, such as model predictive control [1], iterative LQR [2], and, more recently, works related to homotopy methods [3] and iterative optimal control syntheses [4]. When tackling motion planning problems that involve more complicated irregular obstacles, approaches based on an incremental sampling strategy, such as probabilistic roadmaps (RPM) [5] and rapidly-exploring random trees (RRT) [6], can be employed. However, these sampling-based planners are often limited to very simplistic system dynamics, in which any obstacle-free curve between any two points in the state space is automatically dynamically feasible. Further extensions of these methods, such as Kinodynamic RRT* [7] and LQR-RRT* [8] provide the ability to handle kinodynamic system models, which are, however, still limited to either linear or approximately linear dynamics.

In addition to the aforementioned issues, oftentimes, merely avoiding collisions is insufficient in some topology-sensitive tasks, e.g., a vehicle having to stay on the right side of the road. Hence, it is important to consider and distinguish among different homotopy classes for generated trajectories. For this consideration, [9] provides a way to classify homotopy classes in higher-dimensional space and proposes a search-based robot path planning method fulfilling topological constraints. Probabilistic roadmaps introduced in [10] are also capable of path generation under homotopy classes constraints. Furthermore, Gaussian process inference is leveraged in [11] to achieve online motion planning involving multiple homotopy classes. However these methods are again limited to robots with simple dynamics.

There are few prior works on motion planning that allow for the simultaneous considerations of full-scale nonlinear nonholonomic dynamics and homotopy class constraints. Recent works that made important contributions towards this direction are [12], [13] and [14]. These works introduce the so-called affine geometric heat flow, which is a partial differential equation that evolves an arbitrary differentiable path between an initial and final state to a path that meets additional constraints imposed on the problem. A potential drawback is the reliance of this approach on the numerical solution of the involved partial differential equation.

In this paper, we propose a novel direction for addressing motion planning problems with homotopy class constraints that can be applied to general, possibly high-dimensional, nonlinear dynamical systems. The approach first adds an auxiliary control term to the original system, which turns a preset reference trajectory that is dynamically infeasible for the original system to a feasible one for the new extended system. Afterwards, the approach *gradually* (cf. [15],[16]) eliminates the influence of the auxiliary control term so as to let the original input slowly take over the control of the system. As a result, the dynamically infeasible trajectory is gradually deformed to a feasible one that the original system is fully capable to track. In addition, the advocated approach is able to preserve the homotopy class for generated trajectories throughout the iterative synthesis process.

The general idea underlying our proposed technique is reminiscent of the process by which young children first learn how to ride a bicycle, where in the beginning, the children often require external assistance, e.g., via parents actively holding or via a pair of training wheels to mitigate the challenging unstable dynamical component. After having gained sufficient experience under the externally facilitated steady conditions, the external assistance/forces can be gradually removed during subsequent phases of the learning process.

This paper is organized as follows. Section II, introduces the precise problem setup. Then we develop our method for generating feasible trajectories in Section III, and its extension that concerns homotopy class constraints in Section IV. The applicability of the advocated technique is presented in Section V. Finally, we conclude the paper in Section VI.

Department of Electrical and System Engineering, Washington University, St. Louis, MO 63130, USA, {wenbo.he,yunshen.huang,s.zeng}@wustl.edu. This work was supported by the NSF grant CMMI-1933976.

II. Problem Formulation

We consider the general nonlinear dynamical system

$$\dot{x}(t) = f(x(t), u(t)), \tag{1}$$

where $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$. The feasible trajectory of the system above can be represented as a tuple (x, u), which satisfies the dynamical constraint (1) and terminal-point constraints $x(0) = x_{\text{start}}$, $x(T) = x_{\text{target}}$, where T is the terminal time. We denote $x \in \mathbb{X}_{\text{dynamic}}$, if there exists a corresponding admissible input u that steers (1) along x. We take the parallel parking problem as an example to illustrate feasible and infeasible state trajectories as shown in Figure 1.

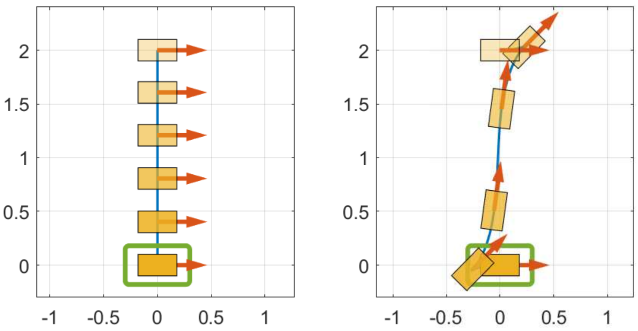


Fig. 1. Parallel parking problem from the current car position $(0, 2)$ to the parking spot that is shown as the green rectangular and centered at $(0, 0)$. The trajectory on the left shown as the blue straight line is dynamically infeasible, which can be observed by the fact that the car's facing directions (shown as the red arrows) are perpendicular to the path. In contrast, the trajectory on the right is feasible, as the car's facing directions are tangent to the path.

Moreover, we denote $\mathcal{H}$ as the homotopy class which satisfies the homotopy class constraints described by a set of obstacles. Trajectories with identical starting and target points are defined to be homotopy equivalent if and only if they can be smoothly deformed into one another without intersecting any obstacle in the state space. A graphical illustration is presented in Figure 2. For a more detailed definition, readers are referred to [9].

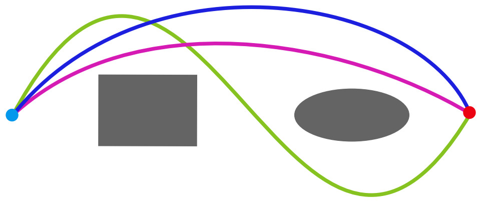


Fig. 2. Three trajectories connecting identical starting and target points. According to the definition of homotopy class, the blue trajectory is homotopy equivalent with the purple one, but belongs to the different homotopy class of the green one.

Given a nominal state trajectory, the objective of this work is to find a feasible trajectory $x \in \mathbb{X}_{\text{dynamic}} \cap \mathcal{H}$ and the corresponding admissible control u, with the property $x \in \mathcal{H}$ being preserved throughout the iterative computation process. It is noted that the initially chosen state trajectories are *not* required to be dynamically feasible.

Notation:

1) For conciseness, $(x_0, x_1, \ldots, x_N)$ will represent the vector $(x_0^\top, x_1^\top, \ldots, x_N^\top)^\top \in \mathbb{R}^{n(N+1)}$.
2) By default, $\|\cdot\|$ stands for $\|\cdot\|_2$ in this paper.

III. Motion Planning by Auxiliary Energy Reduction (AER)

In this section, we first introduce the general structure of the AER method in the absence of a cluttered environment. In Section III-A, we transform the motion planning problem to an energy-minimization problem by virtue of the framework centered around the auxiliary control terms. The resulting nonlinear optimization problem to eliminate the virtually-added control energy is then solved in an iterative fashion, which is described in more detail in Section III-B.

A. Auxiliary control inputs and the Extended System

For the purpose of the computational implementation, we assume that the continuous-time nonlinear system

$$\dot{x}(t) = f(x(t), u(t)), \quad x(t) \in \mathbb{R}^n, \quad u(t) \in \mathbb{R}^m, \tag{2}$$

has been appropriately discretized into

$$x_{k+1} = F(x_k, u_k). \tag{3}$$

In order to track an arbitrary, even dynamically infeasible, nominal trajectory, we introduce an auxiliary (virtual) control component $\hat{u}_k$ to the discrete-time nonlinear system (3)

$$x_{k+1} = F(x_k, u_k) + \hat{u}_k, \tag{4}$$

where $x_k \in \mathbb{R}^n, u_k \in \mathbb{R}^m$, and $\hat{u}_k \in \mathbb{R}^n$. Moreover, for any pair of $X = (x_0, x_1, ..., x_N)$ and $U = (u_0, ..., u_{N-1})$, not necessarily feasible for the original system (3), there always exists a corresponding auxiliary control

$$\hat{u}_k = x_{k+1} - F(x_k, u_k), \tag{5}$$

such that (4) initialized at x_0 and driven with U and $\hat{U}$ traces out the given trajectory X in the state space.

In our Auxiliary Energy Reduction framework, the problem of generating feasible trajectories for the original system can be then cast as an energy minimization problem:

$$\begin{aligned} \underset{(U,\hat{U})}{\text{minimize}} \quad & \|\hat{U}\|^2 \\ \text{subject to} \quad & x_0 = x_{\text{start}}, \\ & x_N = x_{\text{target}}, \\ & x_{k+1} = F(x_k, u_k) + \hat{u}_k, \end{aligned} \tag{6}$$

where the start and end point of the given predetermined trajectory have been fixed. It is clear that if this optimization problem admits a solution where the auxiliary control term reaches zero, i.e., $\|\hat{U}\|^2 = 0$, the virtually-added input no longer impacts the system (3), i.e., we obtain the admissible control that steers the system (3) from x_{start} to x_{target} along with the corresponding feasible trajectory.

The purpose of our overall approach to first introduce an auxiliary term and then to work towards reducing it is to have a feasible starting point (for the extended system) in the computational optimal control setup, which in particular also allows for the fixing of known and desired start and end points. Furthermore, in some instances, a user may already have a sense of how the desired trajectory of a system may end up looking like, and incorporate this information in terms of a useful prior design of the desired state trajectory, with the parts of the prior that are dynamically infeasible for the original system being outsourced to the auxiliary control terms. Another key is that we subsequently solve the nonlinear optimization problem (6) in an iterative fashion which is very efficient in that it results in a sequence of quadratic programs that need to be solved.

B. Auxiliary Energy Reduction

Given the discrete nonlinear system (4), a nominal state trajectory X, a nominal control signal U, and the derived auxiliary input signal $\hat{U}$, we obtain the discrete-time linearization of (4) via perturbing the entire trajectory (except x_0) and control signals, which yields the dynamics of the pertubation of the state as

$$\delta x_{k+1} \approx A_k \delta x_k + B_k \delta u_k + \hat{B}_k \delta \hat{u}_k, \qquad \delta x_0 = 0, \tag{7}$$

where $A_k = \frac{\partial F}{\partial x}(x_k, u_k)$, $B_k = \frac{\partial F}{\partial u}(x_k, u_k)$, $\hat{B}_k = I_n$ are Jacobian matrices of the flow with respect to the state and the input, respectively. Unfolding (7), we arrive at

$$\begin{aligned} \delta x_1 &\approx B_0 \delta u_0 + \hat{B}_0 \delta \hat{u}_0 \\ \delta x_2 &\approx A_1 B_0 \delta u_0 + B_1 \delta u_1 + A_1 \hat{B}_0 \delta \hat{u}_0 + \hat{B}_1 \delta \hat{u}_1 \\ &\vdots \\ \delta x_N &\approx A_{N-1} \dots A_1 B_0 \delta u_0 + \dots + B_{N-1} \delta u_{N-1} \\ &\quad + A_{N-1} \dots A_1 \hat{B}_0 \delta \hat{u}_0 + \dots + \hat{B}_{N-1} \delta \hat{u}_{N-1}. \end{aligned}$$

It is noted that, these linear approximations are only valid for small control perturbations, i.e., $\|\Delta U\| = \|(\delta u_0, \delta u_1, \dots, \delta u_{N-1})\|$ and $\|\Delta \hat{U}\| = \|(\delta \hat{u}_0, \delta \hat{u}_1, \dots, \delta \hat{u}_{N-1})\|$ are sufficiently small. To condense the approximation above, the variation of the entire state trajectory caused by small ΔU and $\Delta \hat{U}$ can be rewritten as

$$\Delta X := \begin{bmatrix} \delta x_1 \\ \vdots \\ \delta x_N \end{bmatrix} \approx \begin{bmatrix} H_1 \Delta U + \hat{H}_1 \Delta \hat{U} \\ \vdots \\ H_N \Delta U + \hat{H}_N \Delta \hat{U} \end{bmatrix} = H \Delta U + \hat{H} \Delta \hat{U}, \tag{8}$$

where

$$H := \begin{bmatrix} B_0 & 0 & \cdots & 0 \\ A_1 B_0 & B_1 & & \vdots \\ \vdots & \vdots & & 0 \\ A_{N-1} \cdots A_1 B_0 & A_{N-1} \cdots A_2 B_1 & \cdots & B_{N-1} \end{bmatrix},$$

$$\hat{H} := \begin{bmatrix} \hat{B}_0 & 0 & \cdots & 0 \\ A_1 \hat{B}_0 & \hat{B}_1 & & \vdots \\ \vdots & \vdots & & 0 \\ A_{N-1} \cdots A_1 \hat{B}_0 & A_{N-1} \cdots A_2 \hat{B}_1 & \cdots & \hat{B}_{N-1} \end{bmatrix},$$

and the matrices H_i and $\hat{H}_i$ represent the ith horizontal block of H and $\hat{H}$, respectively. Moreover, the corresponding state trajectory driven by the slightly drifted nominal control can thus be quantified as

$$X(U + \Delta U, \hat{U} + \Delta \hat{U}) \approx X(U, \hat{U}) + H \Delta U + \hat{H} \Delta \hat{U}. \tag{9}$$

With these insights, we can tackle the solution of the optimization problem (6) in a iterative manner where in each step of the iteration, we are looking at a drastically simpler quadratic program

$$\begin{aligned} \underset{(\Delta U, \Delta \hat{U})}{\text{minimize}} \quad & \|\hat{U} + \Delta \hat{U}\|^2 + \gamma_1 \|\Delta U\|^2 + \gamma_2 \|\Delta \hat{U}\|^2 \\ \text{subject to} \quad & x_N + H_N \Delta U + \hat{H}_N \Delta \hat{U} = x_{\text{target}}, \end{aligned} \tag{10}$$

for gradually reducing the auxiliary energy. Here γ_1, γ_2 are regularization parameters that penalize large values of $\|\Delta U\|$ and $\|\Delta \hat{U}\|$, which ensures the validity of the first order Taylor expansion applied in (8). By iteratively updating U and $\hat{U}$ by solving (10), the auxiliary control energy $\hat{U}$ will be steadily reduced. Furthermore, if we arrive at $\hat{U} = 0$, this would mean that the original input U has taken over full control of the system in steering the state on a trajectory connecting x_{start} and x_{target}. The general structure of AER which is summarized as below.

Algorithm 1 Auxiliary Energy Reduction (AER)

```
Require: A nominal trajectory $X$ connecting $x_{\text{start}}$ and $x_{\text{target}}$, as well as a pair of nominal inputs $(U, \hat{U})$, such that $(U, \hat{U})$ steers the extended system along $X$.
1: Apply the input $(U, \hat{U})$ to the system and calculate $H$, $\hat{H}$.
2: Solve for $(\Delta U^*, \Delta \hat{U}^*)$ of the optimization problem (10).
3: Update the control input via $U \leftarrow U + \Delta U^*$ and $\hat{U} \leftarrow \hat{U} + \Delta \hat{U}^*$.
4: Repeat step $1 - 3$ until $\|\hat{U}\|^2 \leq \epsilon_{2,\text{tol}}$.
```

As the dimension of the input of the extended system $\dim(U) + \dim(\hat{U}) = m + n$ is larger than the dimension of the state, the extended system is an over-actuated system, and more than one pair of $(u_k, \hat{u}_k)$ can drive the system from x_k to x_{k+1}. To decrease the total iteration times needed when employing the AER method, an initial $\hat{U}$ with less energy is preferred. In the next part, we show how this may be achieved by manipulating the way U is initialized.

Without loss of generality, we focus on minimizing the energy of the initial auxiliary control term in the kth time step of the trajectory by considering (5) through choosing the initial input u_k according to $\min\limits_{u_k} ||x_{k+1} - F(x_k, u_k)||^2$. Given a nominal trajectory, a nonlinear regression approach can be utilized to update u_k:

$$u_k \leftarrow u_k + \lambda B_k^{\dagger}(x_{k+1} - F(x_k, u_k)), \tag{11}$$

where λ is the step size which is small enough to ensure a steady convergence, and $B_k^{\dagger}$ is the left pseudo inverse of B_k.

This procedure can be generalized to the entire time steps for properly initialization that is concluded in Algorithm 2.

```
Algorithm 2 Proper initialization of $U, \hat{U}$
Require: A nominal trajectory $X$ from $x_{\text{start}}$ to $x_{\text{target}}$.
1: Initialize $U = (u_0, ..., u_{N-1})$ as a sequence of small random numbers.
2: Calculate $B_k$ for every k, and update $u_k$ by (11) until converged.
3: Set the auxiliary control input via $\hat{u}_k = x_{k+1} - F(x_k, u_k)$, and $\hat{U} = (\hat{u}_0, ..., \hat{u}_{N-1})$.
```

We note that the AER method without optimizing the initial U is fast enough for simpler systems, such that the energy minimization step is not necessary in general, in which case step 2 of Algorithm 2 can be skipped.

IV. Motion Planning with Homotopy Class Constraints

In this section, we illustrate how the AER method can be extended to preserve the homotopy class of generated trajectories when tackling motion planning problems involving cluttered environments and homotopy constraints. To this end, we integrate the AER method with two new components: anchors that are to identify the obstacles and the associated anchor loss that is added to the objective function for maintaining the topological properties of the trajectory.

Anchors are obstacles with regular shapes, such as points, lines and planes. In addition, obstacles with irregular shapes can be approximated by tightly arranging multiple anchors around their boundaries. An anchor is defined as a tuple (C, a), where $C \in \mathbb{R}^{p\times n}$ is the linear operator for dimension selection and rotation, and $a \in \mathbb{R}^p$. As a result, the distance between a point x and the anchor can be measured as $\|Cx - a\|$. Moreover, a set of anchors is defined as a collection of tuples: $\mathcal{A} = \{(C_1, a_1), ..., (C_M, a_M)\}$.

To keep a given trajectory $X(U, \hat{U}) = (x_0, x_1, ..., x_N)$ that is obtained by an extended system (4) starting from x_{start} and driven by inputs $(U, \hat{U})$ away from an anchor set $\mathcal{A}$, we introduce the so-called anchor loss $L(X, \mathcal{A}) \in \mathbb{R}^{M(N+1)}$ by utilizing the framework of barrier functions. Within our particular implementation, the anchor loss is defined to be logarithmic-like, whose $(i + kM)$th element is expressed as

$$L(X, \mathcal{A})_{i+kM} = \begin{cases} 0, & \text{if } \|C_i x_k - a_i\|^2 \geq \mu, \\ -\log(\|C_i x_k - a_i\|^2), & \text{otherwise}, \end{cases} \tag{12}$$

where (C_i, a_i) denotes the ith anchor of $\mathcal{A}$, $\mu \in (0, 1]$ and x_k represents the state at the kth time step of the trajectory. The logarithm distance of the form $-\log(d^2)$ is preferred here because its derivative will not vanish as $-d^2$ or expand too fast as $1/d^2$ when d approaches zero.

In order to prevent the trajectories from crossing through the anchor set during the trajectory updates, we need to ensure that there is no anchor standing inside the smooth deformation from $X(U, \hat{U})$ to $X(U + \Delta U, \hat{U} + \Delta\hat{U})$. To this end, we consider the following. If there is an intersection between (C_i, a_i) and the interpolated trajectory segment x_k to x_{k+1}, we have $\|C_i x_k - a_i\|^2 \leq \|C_i(x_{k+1} - x_k)\|^2$, which implies that there exists x_k, x_{k+1} and an anchor $(C_i, a_i) \in \mathcal{A}$, such that $L(X, \mathcal{A})_{i+kM} \geq -\log(\|C_i(x_{k+1} - x_k)\|^2)$, where we assume that the sampling rate of the discretization from (2) to (3) is sufficiently fast so that $\|C_i(x_{k+1} - x_k)\|^2 \leq \mu$ holds for every k. In addition, by denoting $X^\alpha = X(U + \alpha\Delta U, \hat{U} + \alpha\Delta\hat{U}), \alpha \in [0, 1]$, we can leverage $X^0 \mapsto X^1$ to present a smooth deformation from $X(U, \hat{U})$ to $X(U + \Delta U, \hat{U} + \Delta\hat{U})$. We conclude that, as long as one can ensure

$$\|L(X^\alpha, \mathcal{A})\|_\infty < \min_{i,k} -\log(\|C_i(x_{k+1} - x_k)\|^2) = \Xi, \tag{13}$$

and $X_N^\alpha = X_N^0$ for all $\alpha \in [0, 1]$, the trajectory driven by updated inputs $X(U + \Delta U, \hat{U} + \Delta\hat{U})$ belongs to the same homotopy class of $X(U, \hat{U})$.

It is natural to consider an optimization problem minimizing the norm of the anchor loss so as to fulfill the condition in (13). Before we take the further step towards the optimization formulation, we first investigate the linear approximation of the anchor loss with respect to $(U, \hat{U})$. Based on (8) and (12), the $(i + kM)$th row of the Taylor expansion's first-order term regarding $L(X, \mathcal{A})$ is expressed as

$$\Delta L(X, \mathcal{A})_{i+kM} = \begin{cases} 0, & \text{if } \|C_i x_k - a_i\|^2 \geq \mu, \\ M_{i+kM}\Delta U + \hat{M}_{i+kM}\Delta\hat{U}, & \text{otherwise}, \end{cases}$$

where

$$M_{i+kM} = \frac{-2(C_i x_k - a_i)^T C_i}{\|C_i x_k - a_i\|^2} H_k,$$
$$\hat{M}_{i+kM} = \frac{-2(C_i x_k - a_i)^T C_i}{\|C_i x_k - a_i\|^2} \hat{H}_k.$$

Having done so, the following quadratic program is considered for maintaining the auxiliary energy and decreasing the anchor loss simultaneously,

$$\begin{aligned} \underset{\Delta U}{\text{minimize}} \quad & \|L(X, \mathcal{A}) + M\Delta U\|^2 + \gamma\|\Delta U\|^2 \\ \text{subject to} \quad & x_N + H_N \Delta U = x_{\text{target}}, \end{aligned} \tag{14}$$

where $\gamma \geq 0$ is the regularization parameter.

We note that it is in general cumbersome to choose γ_1 and γ_2 in (10) for each iteration such that the condition (13) is satisfied. Instead, one could simply use the same γ_1 and γ_2 in Algorithm 1, and choose an appropriate update step size α to update the controls: for ΔU^* and $\Delta\hat{U}^*$ obtained from (10), we employ a line search strategy to find the largest update step size $\alpha^* \in (0, 1]$, such that $\|L(X^\alpha, \mathcal{A})\|_\infty < \xi$ for all $\alpha \in [0, \alpha^*]$, where $\xi \leq \Xi$ is a predetermined threshold. This strategy has been found to be particularly effective in dealing with highly nonlinear systems.

To summarize, by incorporating anchors, we endow AER method with the capability of preserving the topology class for the generated trajectories through out iterations. The overall procedure is summarized in Algorithm 3.

Algorithm 3 AER with anchors

```
Require: A nominal trajectory $X$ connecting $x_{\text{start}}$ and $x_{\text{target}}$, as well as a pair of corresponding inputs $(U, \hat{U})$.
1: Apply the input $(U, \hat{U})$ to the system and calculate $H, \hat{H}$.
2: Solve for $\Delta U^*$ and $\Delta \hat{U}^*$ of the optimization problem (10), and find the largest step size $\alpha^*$ using line search.
3: Update $U \leftarrow U + \alpha^* \Delta U$ and $\hat{U} \leftarrow \hat{U} + \alpha^* \Delta \hat{U}$.
4: Calculate $L(X, \mathcal{A})$ and $M$, update $U$ according to (14).
5: Repeat step 4 until $\|L(X, \mathcal{A})\|^2 \leq \epsilon_A$.
6: Repeat step $1 - 5$ until $\|\hat{U}\|^2 \leq \epsilon_{2,\text{tol}}$.
```

V. Illustrative Examples

We first demonstrate the effectiveness of the proposed AER method on two well-studied benchmark platforms: Brockett integrator and a unicycle that operates on a plane. Then a more complex and highly nonlinear model of the quadcopter in 3-D space is evaluated.

A. Brockett Integrator Model

As a classical nonholonomic control system, the Brockett integrator

$$\frac{d}{dt}\begin{pmatrix}x_1\\x_2\\x_3\end{pmatrix}=\begin{pmatrix}u_1\\u_2\\x_1u_2-x_2u_1\end{pmatrix}$$

has been widely studied.

We consider the setup in which $x_0 = (0,0,0)^\top$, $x_{\text{target}} = (0,0,1)^\top$, $T = 2$, and the preset state trajectory is set as a naive straight line between these two points. We set $\gamma_1 = \gamma_2 = 0.05\|\hat{U}\|^2$, and the sampling rate to be 20 Hz. By leveraging the geometric interpretation of the Brockett integrator in terms of the connection to the sector area of the curve in the xy-plane (cf. [17]), the value of x_3 is known to be a function of the area of the (x_1, x_2) curve. Therefore, this preset reference is clearly dynamically infeasible. Figure 3 illustrates the deformation from the reference trajectory to a dynamically feasible one obtained by AER, where the x_0 and x_{target} are represented in red and blue dots, respectively.

B. Unicycle Model with Homotopy Class Constraints

The unicycle model is another popular nonholonomic testbed for illustrating path generation methods, cf. [13]. Its corresponding system dynamics are described by the following system of nonlinear differential equations

$$\frac{d}{dt}\begin{pmatrix}x\\y\\\phi\end{pmatrix}=\begin{pmatrix}v\cos\phi\\v\sin\phi\\\omega\end{pmatrix},$$

where ϕ represents the facing angle of the unicycle, v stands for the moving velocity, and ω denotes the steering velocity.

We set $x_0 = (0,0,0)^\top$, $x_{\text{target}} = (3,0,0)^\top$, $T = 3$. We set $\gamma_1 = \gamma_2 = 0.0001\|\hat{U}\|^2$, $\gamma = 0.005$, and the sampling rate of the discretization to be 20 Hz. In order to illustrate AER method's capability of keeping the consistency of homotopy class for generated trajectories, two anchors are placed at $(1.25, 0.25)$ and $(1.75, -0.25)$, respectively. Figure 4 shows snapshots of the unicycle's movements within three representative iterations, as well as the corresponding input signals.

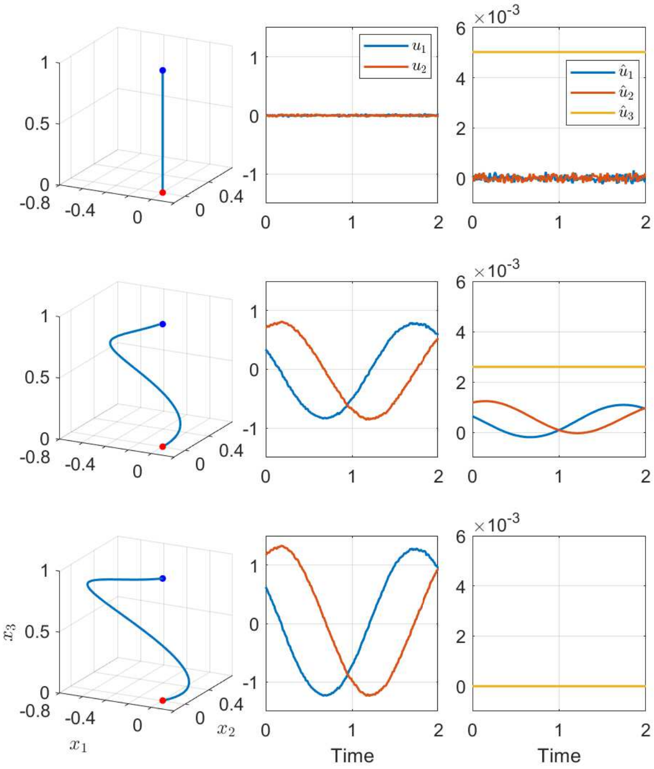


Fig. 3. From top to bottom, the three rows represent the results of the 0th, 40th and 55th iteration of the computation. In the left column, the state trajectory of the Brockett integrator evolves from being dynamically infeasible to being feasible. The corresponding control signals U and auxiliary input signals $\hat{U}$ are shown in the middle and the right column.

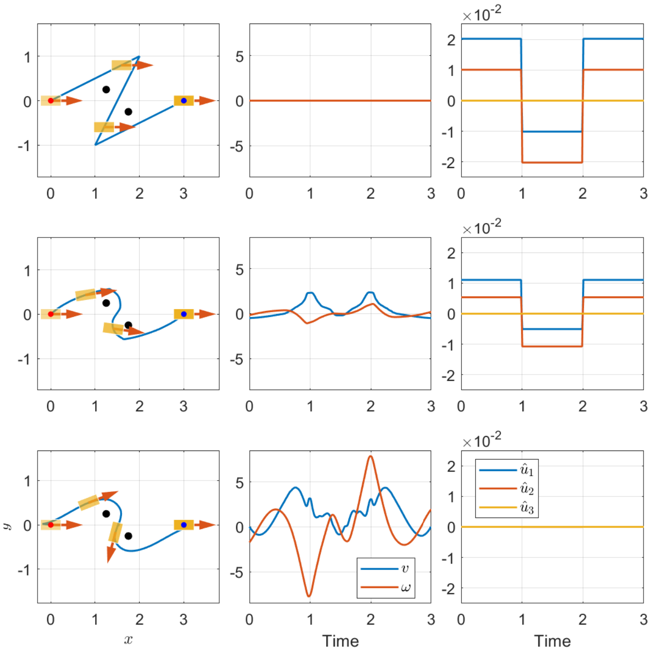


Fig. 4. From top to bottom, the three rows show the computational results of the 0th, 11th and 42th iteration, respectively. In the left column, the generated paths tend to be more dynamically feasible. Additionally, the corresponding control U and the virtually added input $\hat{U}$ are shown in the middle and the right column, respectively.

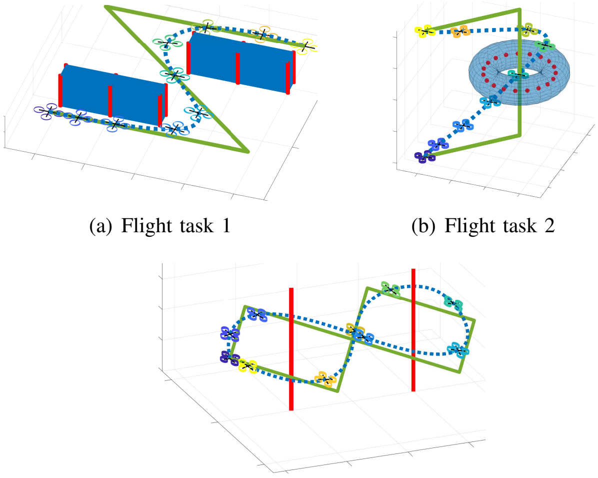
C. Quadcopter Model with Homotopy Class Constraints

Here we consider a quadcopter model with full-scale nonlinear dynamics to highlight the effectiveness of the proposed AER method to generate a feasible tracking trajectory, which at the same time preserves a desired homotopy class. The adopted dynamic model of the quadcopter system is described by a system of nonlinear ODEs with 12 states and 4 control inputs representing the rotating speed of four motors in rotation per minute (rpm) and takes the form

$$\frac{d}{dt}x = f_d(x) + \sum_{i=1}^{4} f_i(x)u_i^2,$$

where f_d and f_i are nonlinear differentiable functions. Readers are referred to [18] for the detailed dynamics structure.

Given the quadcopter model above, we leverage the AER method to plan feasible trajectories with homotopy constraints for three flight tasks in different environments. The simulation setup and results are shown in Figure 5. In these tasks, we set $\gamma_1 = \gamma_2 = 0.0001\|\hat{U}\|^2$, $\gamma = 0.005$, the sampling rate to be 20 Hz, and total time horizon of each task to be 5, 6 and 8 seconds, respectively. Moreover, we heuristically initialize U such that the quadcopter hovers at the starting point, while the initial $\hat{U}$ then would act as the invisible force that drives the quadcopter along the preset trajectory.



(a) Flight task 1

(b) Flight task 2

(c) Flight task 3

Fig. 5. For each flight task, the cluttered environment contains different types of obstacles that are identified by a set of anchors shown in red points/lines. The states with regard to the position of the preset nominal reference trajectory is presented as green polylines, while the rest, i.e. translational velocities, altitudes, angular velocities, are set to zero. AER method has transformed the preset reference to a dynamically-feasible one shown in the blue dashed curve for the quadcopter to track. Meanwhile, the homotopy class constraints are observed to be fulfilled. The sequential snapshots demonstrate the tracking procedure of the quadcopter starting from the darker blue color to lighter yellow color gradually.

VI. Conclusion

In this paper, we introduced a new motion planning technique called auxiliary energy reduction (AER) method. The AER method obtains a dynamically feasible trajectory through minimizing the energy of the auxiliary control term which is artificially added to the original dynamics. By incorporating the so-called anchor and anchor loss, the method further preserves the homotopy class for generated trajectories. We have demonstrated the practicability of the method in three different simulation examples.

References

[1] F. Farshidian, E. Jelavic, A. Satapathy, M. Giftthaler, and J. Buchli, "Real-time motion planning of legged robots: A model predictive control approach," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 577–584.

[2] J. Chen, W. Zhan, and M. Tomizuka, "Autonomous driving motion planning with constrained iterative LQR," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 244–254, 2019.

[3] K. Bergman and D. Axehill, "Combining homotopy methods and numerical optimal control to solve motion planning problems," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 347–354.

[4] M. Vu and S. Zeng, "Iterative optimal control syntheses for nonlinear systems in constrained environments," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 1731–1736.

[5] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[6] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2. IEEE, 2000, pp. 995–1001.

[7] D. J. Webb and J. Van Den Berg, "Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics," in *2013 IEEE international conference on robotics and automation*. IEEE, 2013, pp. 5054–5061.

[8] A. Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano-Perez, "LQR-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 2537–2542.

[9] S. Bhattacharya, M. Likhachev, and V. Kumar, "Topological constraints in search-based robot path planning," *Autonomous Robots*, vol. 33, no. 3, pp. 273–290, 2012.

[10] R. Kala, "Homotopic roadmap generation for robot motion planning," *Journal of Intelligent & Robotic Systems*, vol. 82, no. 3-4, pp. 555–575, 2016.

[11] K. Kolur, S. Chintalapudi, B. Boots, and M. Mukadam, "Online motion planning over multiple homotopy classes with gaussian process inference," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2358–2364.

[12] M. A. Belabbas and S. Liu, "New method for motion planning for non-holonomic systems using partial differential equations," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 4189–4194.

[13] S. Liu, Y. Fan, and M.-A. Belabbas, "Affine geometric heat flow and motion planning for dynamic systems," *IFAC-PapersOnLine*, vol. 52, no. 16, pp. 168–173, 2019.

[14] S. Liu, Y. Fan, and M. Belabbas, "Geometric motion planning for affine control systems with indefinite boundary conditions and free terminal time," *CoRR*, vol. abs/2001.04540, 2020. [Online]. Available: https://arxiv.org/abs/2001.04540

[15] M. Vu and S. Zeng, "Iterative optimal control synthesis for nonlinear switching systems," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 998–1003.

[16] M. Vu, S. Zeng, and H. Fang, "Health-aware battery charging via iterative nonlinear optimal control syntheses," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 12 485–12 490, 2020.

[17] S. Zeng, "Iterative optimal control syntheses illustrated on the Brockett integrator," vol. 52, no. 16. IFAC Symposium on Nonlinear Control Systems, 2019, pp. 138–143.

[18] F. Sabatino, "Quadrotor control: modeling, nonlinear control design, and simulation," (Masters Degree Project, KTH Royal Institute of Technology), 2015.