Warm-Started Physics-Informed Interior Point Methods for Inequality Constrained Optimal Control

Eunbyeol Cho[1] and Jong-Han Kim[1*]

***Abstract*— This paper proposes a Physics-Informed Interior Point Method (PI-IPM) for inequality-constrained optimal control. The method combines a PINN-based indirect solver for the barrier-augmented two-point boundary value problem with feasible-set mappings and an interior point homotopy for inequality path constraints. To improve practical convergence, we introduce a data-driven warm-start strategy by pretraining the network on trajectories generated using Sequential Convex Programming (SCP). The proposed framework is demonstrated on a representative spacecraft deorbit problem, and it was observed that the warm-started PI-IPM reduces solve time relative to a trapezoidal SCP baseline while producing feasible continuous-time trajectories of competitive quality. These results indicate the potential of PI-IPM as a warm-started indirect framework in repeated-solve settings over related problem instances.**

I. Introduction

Inequality-constrained optimal control problems arise in many engineering applications and remain challenging because one seeks trajectories that satisfy nonlinear dynamics, boundary conditions, and path constraints simultaneously [1]. Traditional approaches are broadly categorized into direct and indirect methods. Direct methods transcribe the continuous-time problem into a finite-dimensional nonlinear program and provide a practical framework for handling complex constraints [2]–[4]. Their computational burden, however, can become significant as the problem size increases. Indirect methods apply Pontryagin's Minimum Principle (PMP) [5] to derive necessary conditions for state, costate, and control, yielding a Two-Point Boundary Value Problem (TPBVP) that preserves the continuous-time structure of the original problem [6], [7]. Nevertheless, indirect methods are often sensitive to initialization and are particularly difficult to use in the presence of inequality path constraints.

Recent research combining neural networks and optimal control has developed along several directions. One approach uses supervised learning to approximate optimal trajectories or control policies directly from offline generated solutions [8]. Another line of work uses learning to accelerate existing optimization algorithms by providing informative initial guesses or reduced problem structures for downstream solvers [9]. A third line uses Physics-Informed Neural Networks (PINNs) [10] to solve the optimal control problem itself by embedding the governing equations or PMP-induced optimality conditions into the training objective.

The present work belongs to the third category. In particular, we focus on PINN-based indirect methods that solve the TPBVP arising from PMP. Existing studies have shown the promise of this approach for aerospace optimal control, especially when boundary conditions are enforced analytically within the network structure using Theory of Functional Connections (TFC) [11], [12]. However, robust handling of inequality path constraints remains a major challenge, particularly in indirect formulations where one seeks a continuous-time solution that remains strictly feasible throughout training.

To address this issue, this work proposes an indirect solution framework for inequality-constrained optimal control, termed the Physics-Informed Interior Point Method (PI-IPM). The proposed method solves the barrier-augmented TPBVP using a PINN-based indirect solver, while the interior point structure is used to incorporate inequality constraints through a central-path homotopy. Since the problem becomes numerically ill-conditioned as the barrier parameter approaches zero, we employ an inexact centering step at each stage to efficiently obtain approximate solutions. While the outer structure follows the IPM homotopy, the inner-loop solver at each stage is implemented via a PINN that minimizes PMP residuals while satisfying boundary conditions analytically using TFC-based constrained expressions [12]. This structure yields continuous-time solutions that remain strictly within the feasible set throughout training.

From a practical standpoint, the convergence of a PINN can be slow when initiated from a random state (cold-start). To mitigate this, we introduce a warm-start strategy by pretraining the PINN using a dataset generated via a direct method (SCP). This pretrained model is used to initialize the PI-IPM solver, providing a favorable starting point for the subsequent indirect solve and improving convergence in practice.

A numerical benchmark study is introduced to illustrate the behavior of the proposed formulation on a representative spacecraft deorbit problem motivated by spacecraft trajectory optimization applications [1].

The main contributions of this paper are:

1) A PINN-based indirect solution framework: an indirect solution framework for inequality-constrained optimal

This work was in part supported by the Korea Research Institute for Defense Technology Planning and Advancement (KRIT) under Grant KRIT-CT-22-030 through the Reusable Unmanned Space Vehicle Research Center, funded by the Korea government (Defense Acquisition Program Administration, DAPA), and in part by the Space Challenge Program funded by Korea AeroSpace Administration (KASA), under Grant RS-2025-16063807. Equal support was received from both sources. *(Corresponding author: J.-H. Kim)*

[1]E. Cho and J.-H. Kim are with the Department of Aerospace Engineering and the Program in Aerospace Systems Convergence, Inha University, Incheon, Republic of Korea. J.-H. Kim is also with the Aerospace Systems Research Institute at Inha University. eunbyeeol@inha.edu, jonghank@inha.ac.kr

control that combines a PINN-based TPBVP solver with a barrier-augmented interior point formulation.
2) A feasibility-preserving neural formulation: boundary conditions are enforced through constrained expressions, while inequality constraints are handled through feasible-set mappings.
3) A warm-start strategy and numerical validation: a pretraining method using SCP-generated datasets, enhanced through rotational invariance augmentation, together with demonstration on a spacecraft deorbit problem.

II. PINN for Constrained OCP

This section introduces a physics-informed neural network framework for solving inequality-constrained optimal control problems (OCPs). The problem is reformulated using a barrier-augmented Hamiltonian, and the PMP conditions are embedded into the network loss function using automatic differentiation.

A. Logarithmic Barrier-based Indirect Method

We consider a general OCP with inequality constraints in Bolza form:

$$\begin{aligned} \underset{\mathbf{u}(\cdot)}{\text{minimize}} \quad & \Phi(t_f, \mathbf{x}(t_f)) + \int_{t_0}^{t_f} \mathcal{L}(t, \mathbf{x}(t), \mathbf{u}(t)) dt \\ \text{subject to} \quad & \dot{\mathbf{x}}(t) = f(t, \mathbf{x}(t), \mathbf{u}(t)), \\ & g_j(\mathbf{x}(t), \mathbf{u}(t)) \leq 0, \quad j = 1, \ldots, m, \\ & \mathbf{x}(t_0) = \mathbf{x}_{\text{init}}, \quad \mathbf{x}(t_f) = \mathbf{x}_{\text{des}}, \end{aligned} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{n_\mathbf{x}}$ and $\mathbf{u} \in \mathbb{R}^{n_\mathbf{u}}$ represent the state and control vectors, respectively. The system dynamics are described by f, and g_j represents m scalar inequality constraints.

Equivalently, the cost can be rewritten to explicitly include the inequality constraints through indicator functions:

$$J = \Phi(t_f, \mathbf{x}(t_f)) + \int_{t_0}^{t_f} \Big(\mathcal{L}(t, \mathbf{x}, \mathbf{u}) + \sum_{j=1}^{m} I_-(g_j(\mathbf{x}, \mathbf{u})) \Big) dt, \tag{2}$$

where $I_-(x)$ is zero when $x < 0$ and infinite otherwise. Since this indicator function is non-differentiable and unsuitable for gradient-based methods, it is approximated using a smooth logarithmic barrier function:

$$\hat{I}_-(g_j(\mathbf{x}, \mathbf{u})) = -\tau \log(-g_j(\mathbf{x}, \mathbf{u})), \quad \text{dom } \hat{I}_- = -\mathbb{R}_{++}, \tag{3}$$

where $\tau > 0$ is the barrier parameter controlling the approximation accuracy, and the set $-\mathbb{R}_{++}$ represents the strictly negative orthant.

This results in the barrier-augmented problem:

$$\begin{aligned} \underset{\mathbf{u}(\cdot)}{\text{minimize}} \quad & \Phi(t_f, \mathbf{x}(t_f)) \\ & + \int_{t_0}^{t_f} \Big(\mathcal{L}(t, \mathbf{x}, \mathbf{u}) - \sum_{j=1}^{m} \tau \log(-g_j(\mathbf{x}, \mathbf{u})) \Big) dt \\ \text{subject to} \quad & \dot{\mathbf{x}}(t) = f(t, \mathbf{x}, \mathbf{u}), \\ & \mathbf{x}(t_0) = \mathbf{x}_{\text{init}}, \quad \mathbf{x}(t_f) = \mathbf{x}_{\text{des}}. \end{aligned} \tag{4}$$

An indirect method is employed, deriving the necessary conditions for optimality from Pontryagin's Minimum Principle. The Hamiltonian for the barrier-augmented problem is defined as:

$$H(t, \mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}; \tau) = \mathcal{L} - \sum_{j=1}^{m} \tau \log(-g_j) + \boldsymbol{\lambda}^T f \tag{5}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{n_\mathbf{x}}$ is the costate vector.

The first-order optimality conditions are given by:

$$\dot{\mathbf{x}} = \frac{\partial H}{\partial \boldsymbol{\lambda}}, \quad \dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{x}}, \quad \frac{\partial H}{\partial \mathbf{u}} = 0. \tag{6}$$

These differential equations, combined with the boundary conditions $\mathbf{x}(t_0) = \mathbf{x}_{\text{init}}$ and $\mathbf{x}(t_f) = \mathbf{x}_{\text{des}}$, form a two-point boundary value problem, which is traditionally solved via shooting methods. We propose to model the solution Hamiltonian using a PINN.

B. PINN Training Framework

We adopt the PINN framework proposed in [10] to model the solution of the TPBVP. PINNs embed the governing physical laws given in partial differential equations (PDEs) directly into the neural network's loss function.

In this study, a Multi-Layer Perceptron (MLP) is used to solve the TPBVP, taking time t as an input and yielding the state $\mathbf{x}(t)$, costate $\boldsymbol{\lambda}(t)$, and control $\mathbf{u}(t)$ as outputs. In standard PINN formulations, boundary conditions are typically included in the loss as soft penalties, which may lead to loss imbalance and sensitivity to weighting parameters. To avoid this issue, we enforce the boundary conditions analytically using the *Constrained Expression (CE)* from the Theory of Functional Connections [10]–[12].

Rather than directly using the raw network output $f^{\mathbf{x}}_{\text{NN}}(t)$ as the state, we construct

$$\begin{aligned} \mathbf{x}(t) = f^{\mathbf{x}}_{\text{NN}}(t) &+ \frac{t_f - t}{t_f - t_0} \big(\mathbf{x}_{\text{init}} - f^{\mathbf{x}}_{\text{NN}}(t_0) \big) \\ &+ \frac{t - t_0}{t_f - t_0} \big(\mathbf{x}_{\text{des}} - f^{\mathbf{x}}_{\text{NN}}(t_f) \big), \end{aligned} \tag{7}$$

which analytically satisfies $\mathbf{x}(t_0) = \mathbf{x}_{\text{init}}$ and $\mathbf{x}(t_f) = \mathbf{x}_{\text{des}}$. As a result, the training loss consists only of the PMP residual evaluated at n collocation points $\{t_i\}_{i=1}^{n}$.

$$L_{\text{PMP}} = \frac{1}{n} \sum_{i=1}^{n} \left(\left\| \dot{\mathbf{x}} - \frac{\partial H}{\partial \boldsymbol{\lambda}} \right\|_2^2 + \left\| \dot{\boldsymbol{\lambda}} + \frac{\partial H}{\partial \mathbf{x}} \right\|_2^2 + \left\| \frac{\partial H}{\partial \mathbf{u}} \right\|_2^2 \right), \tag{8}$$

III. Physics-Informed Interior Point Method

A. Algorithm Formulation

To solve the barrier-augmented optimal control problem introduced earlier, we adopt an Interior Point Method (IPM) framework. The core idea of this method is to solve a sequence of unconstrained problems parameterized by a barrier parameter $\tau > 0$, which is progressively reduced. As $\tau \to 0$, the solution of the barrier-augmented problem converges to the original constrained solution. The duality

gap in this context is proportional to $m\tau$, where m is the number of inequality constraints [13].

The sequence of solutions for decreasing values of τ forms a trajectory known as the *central path*. A key insight in modern IPMs is that solving each subproblem precisely, staying on the central path, is inefficient, and an *inexact solution* that remains within a neighborhood of the central path would suffice. This principle motivates our proposed algorithm, which leverages a PINN to efficiently find inexact solutions to the centering subproblems.

In our approach, the PINN is trained to solve the centering problem at each outer iteration. Instead of training the network to full convergence, the optimization is terminated once the loss falls below a prescribed inexactness tolerance, ψ. The trained network parameters from the current step then serve as a warm-start for the next, emulating the behavior of Newton's method in traditional IPMs. Both the barrier parameter τ and the inexactness tolerance ψ are updated according to a geometric decay schedule. Here, $\mu \in (0, 1)$ is the barrier reduction factor that controls the geometric decay of the barrier parameter across outer iterations. Given initial values τ_0, ψ_0 and final targets $\tau_{\text{fin}}, \psi_{\text{fin}}$, the number of updates to the outer loop n_{update} and the decay rate of the tolerance ρ, are determined as follows:

$$\rho = \left(\frac{\psi_{\text{fin}}}{\psi_0}\right)^{\frac{1}{n_{\text{update}}}}, \quad n_{\text{update}} = \left\lceil \frac{\log\left(\tau_{\text{fin}}/\tau_0\right)}{\log(\mu)} \right\rceil. \tag{9}$$

The parameters are then updated at each iteration $k = 0, 1, \ldots$ using the schedule:

$$\tau^{(k+1)} = \mu\tau^{(k)}, \quad \psi^{(k+1)} = \rho\psi^{(k)}. \tag{10}$$

The proposed PI-IPM algorithm begins by initializing the barrier parameter τ, the inexactness tolerance ψ, and the neural network parameters. The algorithm then enters an outer loop where it iteratively solves a series of centering subproblems. In the inner loop, the neural network approximates the state, costate, and control trajectories. The PMP residuals are formulated as a loss function as in (8), which is minimized by the network parameters via gradient descent.

The inner loop terminates when the PMP loss falls below the current inexactness ψ, indicating completion of an *inexact centering step*. Upon completion, the algorithm proceeds to the next outer loop iteration. Here, the barrier parameter τ and the inexactness tolerance ψ are updated according to the decay schedule. The previously trained network parameters serve as a warm-start for the new inner loop. This process repeats until the barrier parameter τ reaches its predefined final value τ_{fin}, which is determined based on the desired duality gap. The final output of the trained network is then taken as the optimal trajectories $\mathbf{x}^*_\tau, \boldsymbol{\lambda}^*_\tau, \mathbf{u}^*_\tau$ corresponding to this terminal barrier parameter.

B. Feasible Set Mapping

Logarithmic barrier functions are well-defined only in the strict interior of the feasible set. Since the raw neural network output is unconstrained, it may violate the inequality constraints during training and thus lead to numerical instability.

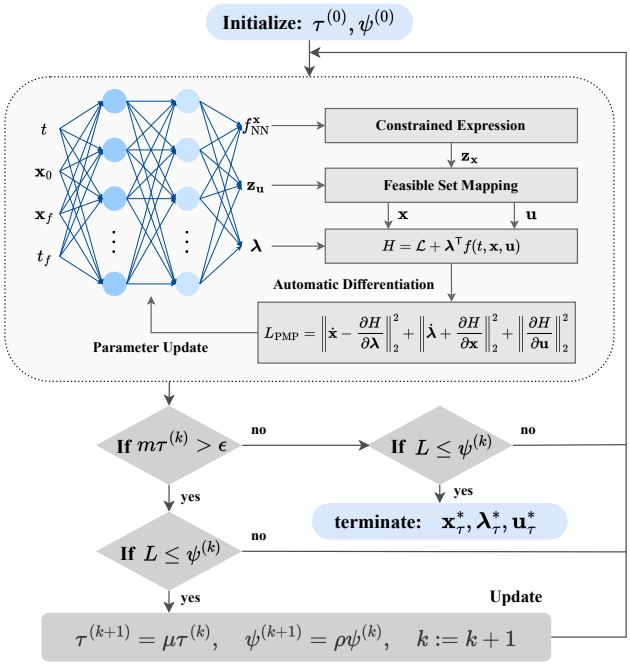

Fig. 1. Schematic of the proposed PI-IPM framework.

To avoid this issue, we apply a feasible-set mapping to the final layer of the network so that the predicted variables satisfy the constraints by construction. For the norm-bounded constraints considered in this paper, the raw output $\mathbf{z}_\theta$ is mapped to a feasible point through smooth transformations. Specifically, the lower- and upper-bound mappings are defined as

$$\mathbf{y}(\mathbf{z}_\theta) = \Big(y_{\text{lb}} + \text{Softplus}(s_\theta)\Big) \frac{\mathbf{z}_\theta}{\|\mathbf{z}_\theta\|_2 + \varepsilon}, \tag{11}$$

and

$$\mathbf{y}(\mathbf{z}_\theta) = \left[\frac{y_{\text{ub}}}{2} \tanh(\|\mathbf{z}_\theta\|_2) + \frac{y_{\text{ub}}}{2}\right] \frac{\mathbf{z}_\theta}{\|\mathbf{z}_\theta\|_2 + \varepsilon}, \tag{12}$$

respectively, where s_θ is an auxiliary scalar output, $\varepsilon > 0$ is a small constant introduced to avoid division by zero, and $\text{Softplus}(x) = \frac{1}{\beta}\log(1 + e^{\beta x})$ is a smooth positive function. Similar mappings can be constructed for other constraint types. With this mapping, the network output remains within the feasible domain throughout training, allowing the barrier-based optimization to proceed stably.

Fig. 1 summarizes the PI-IPM pipeline. The network outputs are wrapped by the constrained expression and the feasible set mapping to enforce the boundary and path constraints. The outer loop then decays the barrier parameter τ and the inexactness tolerance ψ, and warm-starts the next centering step.

C. Warm-Start Strategy

To improve practical convergence of the proposed algorithm, we introduce a pretraining-based warm-start strategy. Because the optimal trajectory is determined by scenario

parameters—such as the initial state $\mathbf{x}_0$, final state $\mathbf{x}_f$, and time of flight t_f—the network architecture is conditioned on these parameters. This design enables the network to learn a direct mapping from problem definition to its corresponding solution trajectory.

The trajectory dataset used for pretraining is generated using Sequential Convex Programming (SCP), which iteratively solves a sequence of convexified subproblems to approximate a non-convex OCP. The resulting pretrained model provides an informed initialization for the PI-IPM solver, reducing the computational load of the subsequent indirect solve. To further enhance generalization across diverse scenarios, the dataset is augmented using rotation invariance techniques, as detailed in Section IV-B.

IV. Numerical Experiments

A. Problem Formulation

To illustrate the proposed framework on a representative inequality-constrained optimal control problem, we consider a benchmark deorbit scenario involving a spacecraft transitioning from a near-circular orbit to atmospheric reentry. The problem assumes planar two-body dynamics with the state vector defined as $\mathbf{x} = \begin{bmatrix}\mathbf{r}^T & \mathbf{v}^T\end{bmatrix}^T$, and seeks to minimize total control energy.

The optimal control problem is formulated as:

$$\begin{aligned}
\underset{\mathbf{u}(\cdot)}{\text{minimize}} \quad & \Phi(t_f, \mathbf{x}(t_f)) + \int_{t_0}^{t_f} \|\mathbf{u}(t)\|_2^2 dt \\
\text{subject to} \quad & \dot{\mathbf{r}}(t) = \mathbf{v}(t), \\
& \dot{\mathbf{v}}(t) = -\frac{\mu_E}{\|\mathbf{r}(t)\|_2^3}\mathbf{r}(t) + \mathbf{u}(t), \\
& \|\mathbf{u}(t)\|_2 \le u_{\max}, \\
& \|\mathbf{r}(t)\|_2 \ge r_{\min}, \\
& \mathbf{x}(t_0) = \mathbf{x}_{\text{init}}, \quad \mathbf{x}(t_f) = \mathbf{x}_{\text{des}},
\end{aligned} \tag{13}$$

where $\mathbf{r} \in \mathbb{R}^3$ and $\mathbf{v} \in \mathbb{R}^3$ are the position and velocity vectors of the spacecraft in the Earth-Centered Inertial (ECI) frame. The control input, $\mathbf{u} \in \mathbb{R}^3$, represents the thrust acceleration, and μ_E is the gravitational parameter of the Earth. The problem includes two inequality constraints. An upper bound on the control magnitude, $u_{\max}$, and a minimum altitude limit, $r_{\min}$, defining a no-fly zone. These inequality constraints are incorporated into the cost functional using logarithmic barrier functions, which results in the following barrier-augmented Lagrangian:

$$\begin{aligned}
\mathcal{L}(t, \mathbf{x}, \mathbf{u}; \tau) = \|\mathbf{u}(t)\|_2^2 - \tau \log(u_{\max} - \|\mathbf{u}(t)\|_2) \\
- \tau \log(\|\mathbf{r}(t)\|_2 - r_{\min}),
\end{aligned} \tag{14}$$

and the corresponding Hamiltonian as follows:

$$H(t, \mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}; \tau) = \mathcal{L} + \boldsymbol{\lambda}_{\mathbf{r}}^T \mathbf{v} + \boldsymbol{\lambda}_{\mathbf{v}}^T \Big(-\frac{\mu_E}{\|\mathbf{r}\|_2^3}\mathbf{r} + \mathbf{u} \Big). \tag{15}$$

The partial derivatives of this Hamiltonian, which are required to formulate the PMP loss in (8), are computed using automatic differentiation.

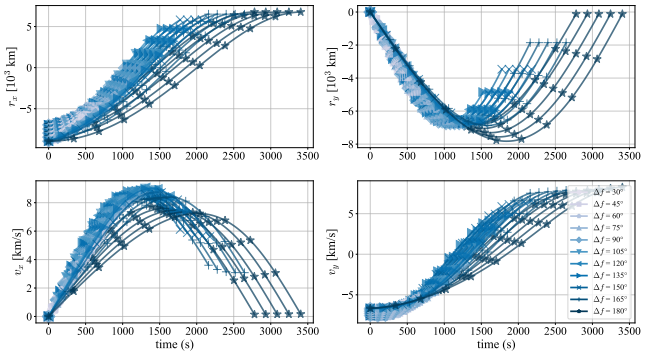

Fig. 2. Sampled dataset: Trajectory (SCP, before augmentation).

B. Data Sampling and Augmentation

Warm-start training data is generated using SCP from sampled coplanar scenarios parameterized by orbital elements $(e, \Delta a, \Delta f)$ as follows:

1) *Eccentricity:* The eccentricity e is drawn from a uniform random distribution over the interval $[0, 0.02]$.
2) *Orbital differences:* The semi-major axis difference Δa and true anomaly difference Δf are sampled deterministically using equi-spaced values.
   - $N_{\Delta a} = 31$ points in $[200, 2600]$ km for Δa.
   - $N_{\Delta f} = 11$ points in $[30°, 180°]$ for Δf.

For each $(\Delta a, \Delta f)$ pair, the initial and the final states are converted to Earth-Centered Inertial (ECI) coordinates, and the Lambert solution is used for initializing the SCP algorithm. This yields a total of $N_{\Delta a} \times N_{\Delta f} = 341$ planar trajectory solutions, as illustrated in Fig. 2.

To improve generalization, the dataset is augmented by exploiting the rotational invariance of two-body dynamics, under which rotations of the coordinate frame preserve the problem structure. Specifically, each planar trajectory is rotated using sampled inclination i, right ascension of the ascending node Ω, and argument of perigee ω to generate valid three-dimensional trajectories without additional SCP solves. This invariance is verified numerically by comparing a directly rotated trajectory with an SCP solution obtained from the correspondingly rotated problem setup, yielding a mean pointwise position error of 3.195×10^{-7} km as in Fig. 3. Using this strategy, the original 341 planar trajectories are expanded to 294,624 three-dimensional trajectories.

C. Pretraining for Warm-Start

The generated trajectory dataset is used for supervised pretraining, yielding the initial model for warm-starting the PI-IPM algorithm. The network is a MLP with two hidden layers of 128 neurons each.

The dataset is split 70% for training and 30% for testing. Table I reports the mean square error (MSE) on both sets, showing strong prediction accuracy and generalization capability of the pretrained model.

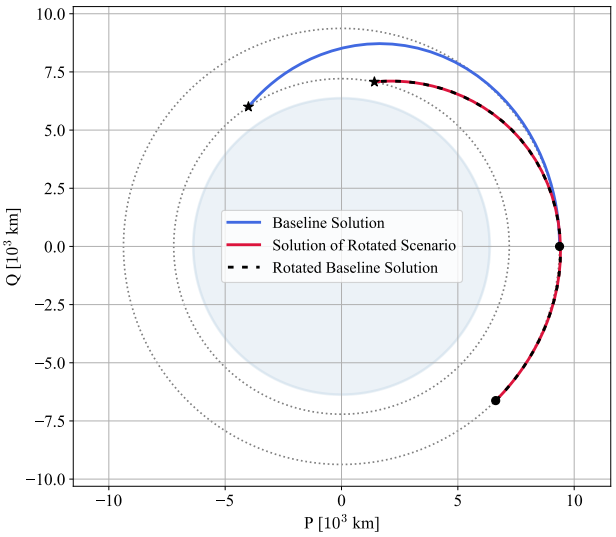

Fig. 3. Verification of rotational invariance.

TABLE I

MEAN SQUARE ERROR OF PRETRAINED MODEL.

Accuracy	$\mathbf{x}$	$\mathbf{u}$	$\boldsymbol{\lambda}$
Train MSE	1.7×10^{-5}	1.3×10^{-6}	8.3×10^{-5}
Test MSE	2.3×10^{-5}	1.3×10^{-6}	9.2×10^{-5}

D. Simulation Results

To evaluate the performance of the proposed PI-IPM algorithm, we conduct a comparative study involving four different approaches:

- Conventional SCP solver with the Euler discretization.
- Conventional SCP solver with the trapezoidal scheme.
- Proposed PI-IPM with random initialization (cold-start).
- PI-IPM initialized by the pretrained model (warm-start).

The physical parameters for the simulation scenario and the nondimensionalization scheme applied to the variables are summarized in Tables II and III, respectively.

The PMP residuals are evaluated using 128 collocation points, and the network parameters are optimized using the Adam optimizer. For IPM scheduling, the barrier parameter is initialized at $\tau_0 = 10^{-3}$ and progressively reduced by a factor of $\mu = 0.1$ until reaching a final value of $\tau_{\text{fin}} = 10^{-7}$. The inexactness tolerance ψ is similarly decreased from $\psi_0 = 10^{-3}$ to $\psi_{\text{fin}} = 10^{-6}$.

TABLE II

PARAMETERS FOR THE SIMULATION SCENARIO.

Parameter	Symbol	Value	Unit
Gravitational parameter	μ_E	398600.4418	km^3/s^2
Earth's radius	R_E	6371	km
Gravitational acceleration	g_0	9.81×10^{-3}	km/s^2
Initial position	$\mathbf{r}_0$	$(9371, 0, 0)$	km
Initial velocity	$\mathbf{v}_0$	$(0, 6.522, 0)$	km/s
Final position	$\mathbf{r}_f$	$(-3755.9, 5633.8, 0)$	km
Final time	t_f	3000	s
Maximum control magnitude	$u_{\max}$	0.55	m/s^2
Minimum orbital radius	$r_{\min}$	6771	km

TABLE III

SCALE FACTORS FOR NONDIMENSIONALIZATION.

Variable	Scale Factor
Time (t)	$\sqrt{R_E/g_0}$
Position ($\mathbf{r}$)	R_E
Velocity ($\mathbf{v}$)	$\sqrt{R_E g_0}$
Control ($\mathbf{u}$)	g_0
Gravitational parameter (μ_E)	$g_0 R_E^2$

TABLE IV

PERFORMANCE COMPARISON OF SCP AND PI-IPM SOLVERS.

Method	Runtime [s]	Position Error [km]	Control [m/s]
SCP (Euler)	12.69	171.7	69.85
SCP (Trapezoidal)	23.05	2.074	53.33
PI-IPM (cold-start)	43.51	5.785	65.28
PI-IPM (warm-start)	**5.048**	**3.794**	**59.46**

All simulations were performed on a desktop equipped with an Intel Core i5-14500 CPU and an NVIDIA GeForce RTX 4060 GPU. The convex subproblems involved in SCP were solved using the MOSEK solver.

Table IV summarizes the performance metrics for all methods. Here, the position error is defined as the trajectory deviation from the reference solution, while the control metric is defined as the cumulative control magnitude. In terms of computation time, the warm-started PI-IPM is the fastest, completing in 5.048 seconds. Regarding solution quality, the SCP method (trapezoidal) achieves the best results. The corresponding control trajectories are shown in Fig. 4. The reduced computation time of the warm-started PI-IPM is attributed to the effectiveness of pretrained initialization, as illustrated in Fig. 5. The cold-start variant begins from random parameters and therefore requires substantially more epochs at each IPM stage, whereas the warm-started model satisfies the inexactness tolerance ψ much earlier.

To further assess the robustness of the proposed method, we additionally consider out-of-distribution scenarios outside the warm-start training range by varying the initial altitude and the corresponding time of flight while keeping the terminal condition fixed. The resulting three-dimensional trajectories are shown in Fig. 6. Even under these unseen conditions, the warm-started PI-IPM generates feasible trajectories that remain close to the reference SCP solutions and consistently converge to the common terminal state, indicating that the pretrained initialization provides a useful prior beyond the nominal training distribution.

Overall, the proposed warm-started PI-IPM provides a favorable trade-off between computation time and solution quality. While the trapezoidal SCP solution remains the highest-quality reference, the proposed method achieves substantially shorter runtime while maintaining consistent trajectory generation, including on out-of-distribution scenarios.

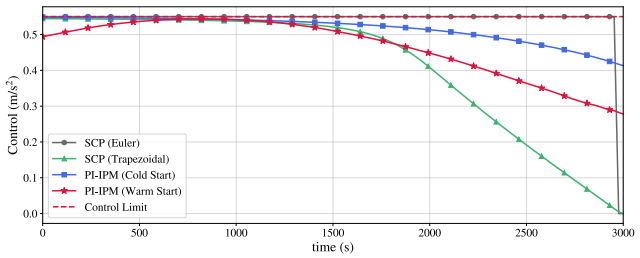

Fig. 4. Control profiles from SCP and PI-IPM solvers.

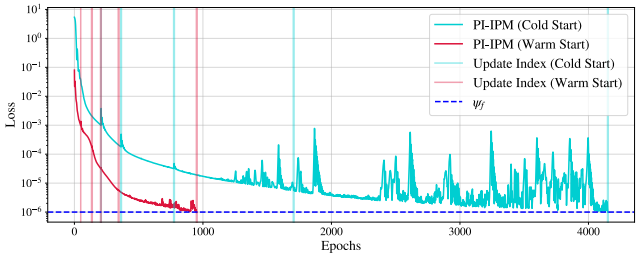

Fig. 5. Convergence comparison of cold-start and warm-start initialization.

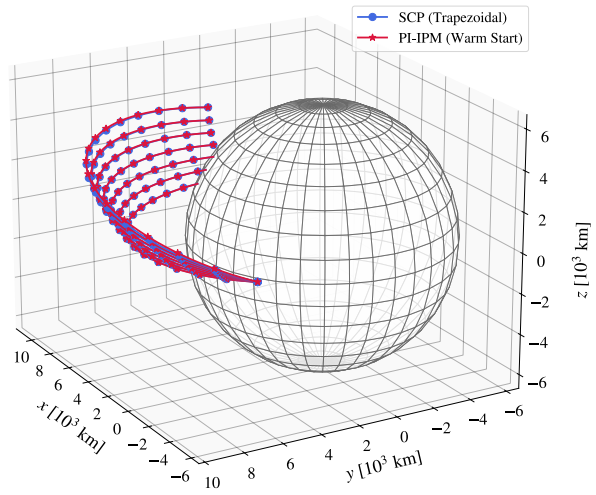

Fig. 6. Representative out-of-distribution trajectories obtained by varying the initial altitude and time of flight while keeping the terminal condition fixed.

As an indirect method, PI-IPM also represents the solution in continuous time, allowing trajectory evaluation at arbitrary resolution for downstream applications such as feedback control or onboard replanning.

V. Conclusion

This paper presented PI-IPM, a PINN-based indirect framework for inequality-constrained optimal control. The method combines a barrier-augmented interior point homotopy with a physics-informed TPBVP solver, while TFC-based constrained expressions and feasible-set mappings are used to enforce boundary conditions and maintain feasibility during training.

Simulation results validated the effectiveness of the proposed approach. In particular, the data-driven warm-start strategy, constructed from Sequential Convex Programming (SCP) and augmented via rotational invariance, achieved a reduction in computation time of approximately 78% relative to a high-accuracy SCP (trapezoidal) baseline. This improvement is attributed to the pretrained neural network providing high-quality initializations, which substantially reduce the number of iterations required for online optimization.

The reduction in computation time, coupled with competitive solution accuracy (mean trajectory deviation of 0.321%), demonstrates the potential of the proposed method for real-time or large-scale applications. These include rapid trajectory replanning under uncertainty, onboard optimization, or Monte Carlo simulations over broad scenario spaces.

Future research directions include reducing reliance on supervised pretraining data through reinforcement learning or unsupervised techniques, as well as extending the method to more complex mission profiles, such as multi-phase or interplanetary trajectories, where nonlinearities and constraints are even more prominent.

References

[1] A. Shirazi, J. Ceberio, and J. A. Lozano, "Spacecraft trajectory optimization: A review of models, objectives, approaches and solutions," *Progress in Aerospace Sciences*, vol. 102, pp. 76–98, Oct. 2018.

[2] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

[3] L. Biegler and V. Zavala, "Large-scale nonlinear programming using IPOPT: An integrating framework for enterprise-wide dynamic optimization," *Computers & Chemical Engineering*, vol. 33, no. 3, pp. 575–582, 2009, selected Papers from the 17th European Symposium on Computer Aided Process Engineering held in Bucharest, Romania, May 2007.

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

[5] L. Pontryagin, *Mathematical Theory of Optimal Processes*, ser. Classics of Soviet Mathematics. Taylor & Francis, 1987.

[6] M. Osborne, "On shooting methods for boundary value problems," *Journal of Mathematical Analysis and Applications*, vol. 27, no. 2, pp. 417–433, 1969.

[7] D. D. Morrison, J. D. Riley, and J. F. Zancanaro, "Multiple shooting method for two-point boundary value problems," *Communications of the ACM*, vol. 5, no. 12, pp. 613–614, 1962.

[8] C. Sánchez-Sánchez and D. Izzo, "Real-time optimal control via deep neural networks: Study on landing problems," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 5, pp. 1122–1135, 2018.

[9] J. Briden, T. Gurga, B. J. Johnson, A. Cauligi, and R. Linares, "Tight constraint prediction of six-degree-of-freedom transformer-based powered descent guidance," in *AIAA SCITECH 2025 Forum*, 2025, pp. 2025–2774.

[10] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[11] E. Schiassi, A. D'Ambrosio, H. Johnston, M. De Florio, K. Drozd, R. Furfaro, F. Curti, and D. Mortari, "Physics-informed extreme theory of functional connections applied to optimal orbit transfer," in *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference, Lake Tahoe, CA, USA*, 2020, pp. 9–13.

[12] E. Schiassi, A. D'Ambrosio, K. Drozd, F. Curti, and R. Furfaro, "Physics-informed neural networks for optimal planar orbit transfers," *Journal of Spacecraft and Rockets*, vol. 59, 01 2022.

[13] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.