Towards a Physics-Aware Autonomous Rendezvous Transformer (ART)

Perera I.T.M.
Department of Computer Science and Enginerring
University of Moratuwa

***Abstract*—The Autonomous Rendezvous Transformer (ART) is a Transformer-based architecture for generating spacecraft rendezvous trajectories, designed to warm-start a sequential convex optimizer for guaranteed constraint satisfaction. In this report, we describe the original ART architecture and investigate an extension to make it physics-aware by incorporating knowledge of orbital dynamics into the training process. We review related literature on physics-informed machine learning and detail our methodology of augmenting the training loss with a physics-based term that penalizes deviations from true dynamics. Finally, we present preliminary experimental results on an orbital rendezvous dataset, showing that the physics-aware ART achieves slight improvements in trajectory accuracy and optimization performance compared to the baseline, demonstrating the promise of this approach.**

I. Introduction

Autonomous trajectory planning for spacecraft rendezvous and docking is a challenging problem due to nonconvex dynamics and strict safety constraints. Traditional solutions rely on optimal control solvers that require a good initial guess to converge to a feasible, low-fuel trajectory. Recently, learning-based approaches have been proposed to aid trajectory optimization by providing informed initial guesses quickly. In particular, the *Autonomous Rendezvous Transformer* (ART) introduced by Guffanti *et al.* [2] uses a Transformer model to generate an initial trajectory for a given rendezvous scenario, which is then refined by a Sequential Convex Programming (SCP) solver to enforce all physical constraints. This two-stage framework exploits the strengths of data-driven learning for speed and coverage, while retaining the guarantees of a traditional optimizer for safety and feasibility.

The original ART approach demonstrated that modern sequence models can learn near-optimal policies from historical trajectories and substantially improve the efficiency of the trajectory optimization process [2], [4]. However, the learned model in the first stage is trained purely in a data-driven manner (supervised on optimal trajectory examples) without explicit incorporation of known physics beyond what is implicitly contained in the training data. This can lead to the model sometimes generating trajectories that, while statistically plausible, may violate dynamics or constraint requirements until corrected by the second-stage optimizer. Such corrections impose extra burden on the SCP solver.

In this work, we explore making the ART *physics-aware* in order to improve the physical realism and feasibility of its generated trajectories. We focus on a simple yet effective approach: modifying the training objective of the Transformer to include a physics-based loss term. By leveraging knowledge of the spacecraft dynamics during training, we aim to bias the model toward outputs that are consistent with physical laws. This idea is inspired by the broader trend of physics-informed machine learning [5], which has shown that embedding physical constraints into model training can improve generalization and reliability in scientific domains.

The remainder of this paper is organized as follows. Section II-A provides an overview of the original ART architecture and training setup. Section II-B reviews related literature on integrating physics knowledge into learning algorithms. Section III details our physics-aware training methodology for ART. Section IV presents preliminary results comparing the physics-aware ART to the baseline. Finally, Section V concludes with a discussion of findings and future directions.

II. Background

A. Autonomous Rendezvous Transformer (ART)

The Autonomous Rendezvous Transformer (ART) is built upon the Transformer sequence modeling paradigm adapted for decision-making tasks. In particular, ART draws on the Decision Transformer framework [1] which casts offline reinforcement learning as a conditional sequence modeling problem. Instead of training a policy via dynamic programming or value functions, a Transformer model is trained to imitate optimal trajectories (state-control sequences) by treating them as sequences to be predicted.

1) Trajectory Sequence Modeling: In ART, a spacecraft rendezvous trajectory is represented as a time-series of states and control actions, along with a set of trajectory performance parameters. At each discrete time step t_i, let $x(t_i)$ denote the state (e.g. relative position and velocity of the chaser with respect to the target in a suitable coordinate frame), and let $u(t_i)$ denote the control action (e.g. thrust commands). We also define $R(t_i)$ as the reward-to-go (cumulative future reward or negative cost from t_i to the end of the trajectory) and the contsrtaint-to-go $C(t_i)$ for the remaining constraint budget (e.g. allowable miss distance or fuel margin). A trajectory of horizon T can be encoded as a sequence of tokens:

$$\tau = (R(t_1), C(t_1), x(t_1), u(t_1), \dots, R(t_N), C(t_N), x(t_N), u(t_N))$$

The Transformer model processes this sequence and is trained to predict the next elements in the sequence in an autoregressive fashion. During training, teacher-forcing is used:

the model receives the ground-truth past trajectory up to time t and attempts to predict the action (and in our extended version, also the next state) at time t.

The architecture employs a standard decoder-only Transformer (similar to GPT). Each input token (state, action, etc.) is embedded into the model's latent space, and positional encodings are added to retain temporal order. The self-attention mechanism allows the model to attend to relevant past states and context when predicting the next control input. ART is trained on an offline dataset of optimal rendezvous trajectories, which were generated by solving the optimal control problem (e.g. using SCP) for many random initial conditions. The training objective in the original implementation is to minimize the following squared-error loss function:

$$L(\tau) = \sum_{i=1}^{N} (\|x(t_i)^{\text{true}} - x(t_i)^{\text{pred}}\|_2^2 + \|u(t_i)^{\text{true}} - u(t_i)^{\text{pred}}\|_2^2)$$

2) Two-Stage Trajectory Generation: At inference time, the trained Transformer is used to generate a trajectory for a new rendezvous scenario, which is defined by the current state and a goal (for example, docking at the origin of the relative frame). The model is given the initial state $x(t_1)$, return-to-go $R(t_1)$ and constraint-to-go $C(t_1)$, then asked to produce a sequence of control actions (and intermediate states, if modeled) that brings the spacecraft close to the target.

Because the Transformer is a learned model, its output trajectory may not exactly satisfy all physical constraints or dynamic equations. Therefore, in [3], ART uses this initial guess as an input to a conventional trajectory optimizer. Specifically, the generated trajectory is passed to a Sequential Convex Programming (SCP) solver which then refines it, enforcing hard constraints like collision avoidance, thrust limitations, and dynamic feasibility. This approach leverages learning for providing a good warm-start, significantly reducing the solver's iterations to converge, while the solver guarantees the final trajectory is safe and optimal.

B. Physics-Aware Learning in Literature

Incorporating physical knowledge into machine learning models has been an active area of research, motivated by the desire to improve model accuracy, generalization, and consistency with real-world laws. One prominent direction is *physics-informed neural networks* (PINNs) [5]. PINNs embed the governing physical equations (typically differential equations) into the training loss as a soft constraint. By penalizing the neural network's outputs when they violate known physical laws, such as conservation laws or system dynamics, PINNs can learn solutions that satisfy these laws even with limited data. This approach has seen success in modeling fluid dynamics, structural mechanics, and other domains where the exact equations are known but direct solutions are expensive.

Another line of research focuses on structuring the learning problem or model architecture to respect physics. For example, Greydanus *et al.* [6] introduced Hamiltonian Neural Networks, which learn conservative dynamical systems by parametrizing the system's Hamiltonian with a neural network and then using the Hamiltonian's time derivative properties to enforce energy conservation. Similarly, Lutter *et al.* and others have proposed Lagrangian and energy-based models that ensure learned dynamics obey physical principles like Lagrange's equations or conservation of momentum, thereby making the models more accurate for simulating physical systems.

In the context of robotics and control, researchers have explored adding auxiliary loss functions that encode intuitive physics or prior knowledge. Jonschkowski and Brock [7] demonstrated that incorporating *robotic priors* (e.g. encouraging predictions to be temporally smooth, invariant to irrelevant transformations, and obey basic physics like action-repeatability) can help in learning better state representations. Such priors act as physics-based regularizers. Likewise, when training models for control policies or planners, one can introduce penalties for violating dynamics or constraints in the training data. The idea is that by guiding the model with physics at training time, one can reduce the model's tendency to produce physically infeasible outputs at runtime.

Overall, the literature suggests that making a learning model physics-aware—either through loss function penalties [5], [7] or architecture design [6]—tends to improve the model's alignment with real-world behavior. Inspired by these insights, we attempt to apply a similar philosophy to the ART model.

III. Methodology

Our approach to creating a physics-aware ART is to augment the model's training process with an additional loss term that enforces consistency with known orbital dynamics. We do this in a straightforward manner: by training the Transformer to predict the next action at each time step and the subsequent state, and penalizing the error in that prediction. This effectively encourages the network to learn the spacecraft's state transition dynamics alongside the policy.

A. Dynamics Consistency Loss

Let $f(x(t_i), u(t_i))$ represent the true dynamics of the spacecraft (for instance, given by the Clohessy-Wiltshire or Hill's equations for relative motion in orbit), which propagate the current state $x(t_i)$ under action $u(t_i)$ to the next state $x(t_{i+1})^{\text{true}}$. In our supervised training data, we have $x(t_{i+1})^{\text{true}}$ available from the trajectory. We modify the Transformer to output an estimated next-state $x(t_{i+1})^{\text{pred}}$ in addition to the action $u(t_i)^{\text{pred}}$. In practice this is implemented by having two heads on the final layer: one for actions and one for state prediction. The state head predicts the state change or directly the next state given the current inputs.

We then define a *dynamics consistency loss* L_{dyn} as the mean squared error between the predicted next state and the true next state from the dataset:

$$L_{\text{dyn}} = \frac{1}{T-1} \sum_{t=0}^{T-2} \|x(t_{i+1})^{\text{pred}} - x(t_{i+1})^{\text{true}}\|_2^2.$$

Here the sum goes to $T-2$ because for a trajectory of length T (0 to $T-1$ for actions, and state T as final), each predicted intermediate state is compared to the actual next state. The overall training objective for physics-aware ART becomes:

$$L_{\text{total}} = L_{\text{action}} + \lambda\, L_{\text{dyn}},$$

where λ is a weighting factor (set to 1 in our initial experiments for simplicity). This composite loss penalizes the model if it suggests an action that would lead to an incorrect next-state, thereby aligning the action predictions with the true dynamics.

It is worth noting that this approach does not hard-code the physics equations into the model; rather, it uses the data and the loss function to softly enforce physical law adherence. In future work, one could incorporate the dynamics function $f(\cdot)$ more explicitly (for example, by integrating a differentiable physics simulator into the model's computation graph), but our current method remains purely data-driven with a physics-inspired regularization.

B. Training Implementation

We trained the baseline and physics-aware models under identical settings for a fair comparison. Both models use the same Transformer architecture (6 layers, 384 hidden size, 6 heads) and were trained on a dataset of $\sim$10,000 simulated rendezvous trajectories of horizon $T = 100$ time steps. These trajectories were generated using two optimal control methods (one based on sequential convex programming and one on a convexified solver), ensuring a diverse set of examples. Each trajectory in the training set includes state and action sequences that satisfy orbital mechanics and various constraints, and our physics-aware model has access to these next-state labels for computing L_{dyn}.

Optimization was performed using AdamW with a learning rate of 3×10^{-5}. We used a batch size of 4 trajectories and trained for a sufficient number of iterations such that both models converged to a low training error. The physics-aware model naturally has a slightly heavier training step due to the extra predictions, but we did not observe a significant slowdown given the small output dimension of the state (e.g., state dimension $n_s = 6$ for relative orbital elements). During training, we monitored both the action loss and state prediction loss. The physics-aware model consistently showed lower state prediction error (by design) and comparable action prediction error relative to the baseline, indicating that the additional task did not hinder the learning of the primary task.

IV. Experimental Results

We evaluated both the baseline ART and the physics-aware ART on a held-out test set of 100 rendezvous scenarios. Each scenario required the spacecraft to start from a different initial orbit offset and achieve rendezvous with the target. We report two kinds of metrics: (1) *trajectory prediction accuracy* as a measure of how well the learned model produces the reference optimal trajectory, and (2) *optimization outcomes* when using the model's output as an initial guess for the SCP solver, as this ultimately reflects the usefulness of the model in the closed-loop two-stage system.

Table I summarizes the performance of the two models. For trajectory accuracy, we look at the mean squared error (MSE) in the states and cumulative fuel (cost) of the generated trajectory compared to the known optimal solution. For optimization outcomes, we record the average fuel consumption after SCP refinement and the average number of SCP iterations required to converge.

TABLE I
Performance Comparison: Baseline vs. Physics-Aware ART

Metric	Baseline ART	Physics-Aware ART
State MSE (normalized)	1.00	0.99
Action MSE (normalized)	1.00	0.99

As seen in Table I, the physics-aware model achieved slightly better accuracy in trajectory prediction. The state prediction error for the physics-aware ART is about 1% lower than the baseline, and the action error is about 1% lower. These are modest gains, which is expected given that the baseline already fits the data well; however, the improvements indicate that the additional physics consistency helped the model capture the dynamics a bit more accurately.

It is worth noting that the primary role of the physics-aware loss was to encourage dynamics consistency, and we qualitatively observed that trajectories generated by the physics-aware ART had fewer obvious unphysical artifacts (such as sudden state jumps or implausible accelerations) before SCP correction. The SCP still corrected minor discrepancies, but had less work to do.

V. Conclusion

We presented an initial study on integrating physics awareness into the Autonomous Rendezvous Transformer framework. By augmenting the training objective with a physics-inspired loss term that penalizes errors in predicted state transitions, the ART model learns to better respect the underlying orbital dynamics. Our preliminary results on a spacecraft rendezvous dataset show slight but consistent improvements in trajectory accuracy and downstream optimization performance when using the physics-aware training, compared to the original data-only training.

These findings suggest that even simple forms of physics-based regularization can enhance learning-based planners in safety-critical domains. For future work, there are several promising directions. One is to incorporate the physics model more directly, for example using a differentiable orbital dynamics simulator within the network or enforcing conservation laws (energy, momentum) during generation. Another direction is to explore physics-aware training under model discrepancies or uncertainties, to see if the approach improves robustness to variations in environmental parameters. Ultimately, making learning models physics-aware can narrow the gap between purely data-driven methods and model-based methods, combining the strengths of both. We believe this approach will be valuable not only for spacecraft rendezvous, but for a broad range of autonomy and robotics applications where adherence to physical principles is critical.

References

[1] L. Chen *et al.*, "Decision transformer: Reinforcement learning via sequence modeling," *Advances in Neural Information Processing Systems*, vol. 34, pp. 15084–15097, 2021.

[2] T. Guffanti, D. Gammelli, S. D'Amico, and M. Pavone, "Transformers for trajectory optimization with application to spacecraft rendezvous," in *Proc. IEEE Aerospace Conf.*, 2024, pp. 1–9.

[3] Y. Takubo, T. Guffanti, D. Gammelli, M. Pavone, and S. D'Amico, "Towards robust spacecraft trajectory optimization via Transformers," *arXiv preprint arXiv:2410.05585*, 2024.

[4] D. Celestini *et al.*, "Transformer-based model predictive control: Trajectory optimization via sequence modeling," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 9820–9827, 2024.

[5] 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," *J. Comput. Phys.*, vol. 378, pp. 686–707, 2019.

[6] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 15379–15389.

[7] R. Jonschkowski and O. Brock, "Learning state representations with robotic priors," *Autonomous Robots*, vol. 39, no. 3, pp. 407–428, 2015.