K-31

強化学習を用いた連続低推力による惑星間遷移のロバスト軌道設計

Robust Interplanetary Trajectory Design with Continuous Low Thrust Using Reinforcement Learning

○深澤和貴[1]，内山賢治[2]，増田開[2]

*Kazuki Fukazawa[1], Kenji Uchiyama[2], Kai Masuda[2]

This paper investigates the use of reinforcement learning for the robust interplanetary trajectory design with continuous low thrust trajectory in presence of missed thrust event. An open-source reinforcement learning algorithm is adopted, and a novel method for the reward function, consisting of an exponential polynomial, is adopted for the training process. To validate the proposed reward function, the guidance laws obtained from the training are compared under scenarios with thrust loss and without thrust loss. Then the corresponding numerical results are presented. The numerical results verify the proposed method effectively handles constraint conditions, achieving a performance comparable to that under scenarios without thrust loss, even under scenarios with thrust loss.

１．緒言

宇宙機の軌道最適化手法は，主に直接法と間接法に分類される．これらの手法は，ミッション中に起こりうるスラスタの故障や軌道推定誤差といった不確実性を考慮しない決定論的なアプローチであり，確率的最適制御問題に対しては適切でない場合が多い．

これに対し，近年では，模倣学習や強化学習(RL)などの機械学習を用いることが注目されている．模倣学習では，RL と比較して計算コストが少ない反面，あらかじめ設計済みの解を利用して学習するため，解の多様性が制限される問題がある．特に，模倣する軌道設計が限定的な環境下でしか有効でない場合には，汎用性が低い．一方，RL ではエージェントが自ら状態空間を広範囲に探索するため，より汎用的で未知の環境にも適応可能な解が得られるという利点がある．しかし，RL では，制約条件をペナルティとして報酬に組み込む必要があるため，制約処理能力がペナルティの設計に依存する．制約条件を厳密に守る必要がある宇宙機の軌道設計では，このことが課題となる．

本研究では，制約条件を指数関数の多項式として報酬に組み込む報酬シェーピングを提案する．さらに，提案手法で訓練されたエージェントが，推力損失下での制約処理において有効であることを数値シミュレーションにより検証する．

２．運動の定式化と実装

以下に，宇宙機の運動を記述する．なお，$\mathbf{r}$は太陽中心の慣性座標系に対する位置$\mathbf{r} = [x \quad y \quad z]^{\mathrm{T}}$である．

$$\ddot{\mathbf{r}} = -\frac{\mu_{\odot}\mathbf{r}}{\|\mathbf{r}\|^3} + \frac{F}{m}\mathbf{R} \tag{1}$$

$$\dot{m} = -c \tag{2}$$

ここで，エージェントにより決定される推力の大きさをF[N],軌道接線座標における推力の方位角と仰角をα[rad], β[rad]とするとき，$\mathbf{R}$は推力の慣性軸方向成分であり，以下の式であらわされる．

$$\mathbf{R} = [\cos\alpha\cos\beta \quad \sin\alpha\cos\beta \quad \sin\beta]^{\mathrm{T}} \tag{3}$$

また，$\mu_{\odot}$は日心重力定数，mは宇宙機の質量[kg]，cは単位時間あたりの燃料消費量[kg/s]である．宇宙機は１ステップで等時点１区間を伝播し，エピソードの終端において以下の制約誤差R_{err}を評価する．なお，$\mathbf{r_{ref}}, \mathbf{v_{ref}}$は拘束条件を与える終端の位置と速度である．

$$R_{err} = \max\left(\frac{\|\mathbf{r_{ref}} - \mathbf{r}\|}{\|\mathbf{r}\|}, \ \frac{\|\mathbf{v_{ref}} - \mathbf{v}\|}{\|\mathbf{v}\|}\right) \tag{4}$$

各エピソードの実行時点で，推力損失を発生させる区間$i \in \mathbb{Z} \cap [0, N)$を，乱数生成により無作為に決定する．エージェントにより決定される入力は以下である．

$$\mathbf{S} = \begin{cases} [F \quad \alpha \quad \beta]^{\mathrm{T}} & (k \neq i) \\ \mathbf{0} & (k = i) \end{cases}$$

本研究では，エピソードの終端$(k = N)$および過程$(k < N)$において，それぞれ以下の報酬関数を用いる．なお，式中で用いた定数の値を **Table 1** に示す．

$$r_k = \begin{cases} -h - jR_{err} + \sum_{n=1}^{I} c_n \exp(-w_n R_{err}) & (k = N) \\ -\int_{t_{k-1}}^{t_k} c\, dt & (k < N) \end{cases} \tag{5}$$

また，用いたハイパーパラメータを **Table 2** に示す．なお，学習率αについては以下の式に従う．ここで，Tは学習を行う全ステップ数であり，tは各時点における経過ステップ数である．

$$\alpha = \alpha_{\infty} + (\alpha_0 - \alpha_{\infty})\exp\left[-d_{rate}\left(1 - \frac{t}{T}\right)\right] \tag{6}$$

１：日大理工・院（前）・航宇　２：日大理工・教員・航宇

Table 1. Reward Function Parameters

Symbol	Value	Symbol	Value
h	12	w_1	0.005
j	10	w_2	0.0005
c_1	1	w_3	1
c_2	10	w_4	10
c_3	12	w_5	50
c_4	7	N	85
c_5	2	I	5

Table 2. PPO Hyperparameters

Parameter	Symbol	Value
Initial Learning Rate	α_0	1×10^{-3}
Final Learning Rate	α_∞	1×10^{-6}
Decay Rate	d_{rate}	6
Initial Clip Range	ϵ_0	0.3
Ent Coef	c_2	1×10^{-6}
GAE Factor	λ	0.99
Discount Factor	γ	0.98

３．数値シミュレーション

本研究では，半径 1[AU]の地球軌道から，半径 1.52[AU] の火星軌道に遷移する問題を扱う．また，宇宙機は初期質量 3000[kg]，比推力 3000[s]，最大出力 1.5[N]とした．以上の環境において，StableBaselines3 の PPO[3]を用いて35×10^6ステップ学習を行い，学習が完了したポリシーを用いて2×10^4ステップシミュレーションを行った．その結果，終端での制約条件を **Table 3** に示す値で満たすことができた．なお，表中の SR とは，制約を推力損失がない状態で達成した制約値以下で満たせた割合である．

また，**Figure 1** では，非推力損失下での軌道を水色で示し，その際の入力系列に対して推力損失を発生させた場合の軌道を灰色，学習済みのポリシーを適用した推力損失下での軌道を青色で示した．結果より，学習済みポリシーでは終端においてランデブー条件を満たせていることがわかる．また，**Figure 2** は，学習の進行に対するR_{err}の値の推移である．学習の進行とともにR_{err}が減少していくことがわかる．

Table 3. R_{err}Value

	SR[%]	Max[10^{-3}]	Min[10^{-3}]
ProposedMethod	71.43	27.82	0.123
Unperturbed	100	4.384	

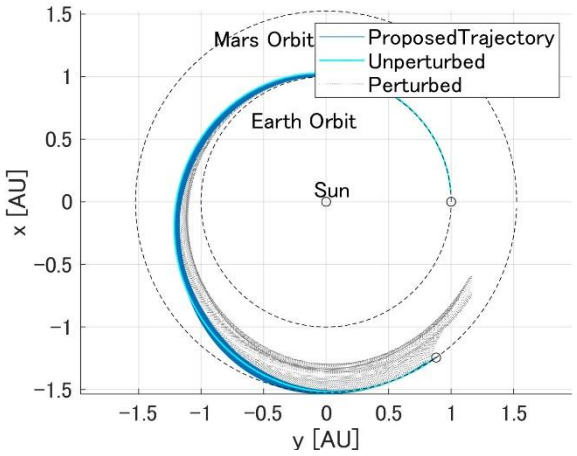

Figure 1. Earth-Mars Trajectories by Robust Policy

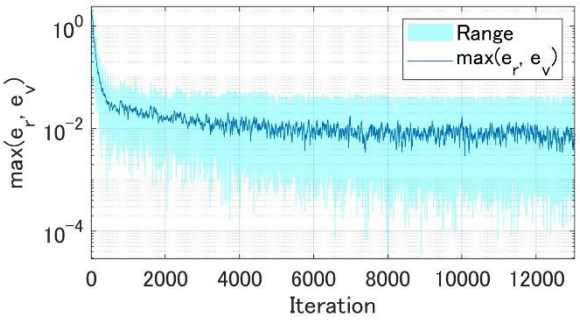

Figure 2. Iteration vs Constraint Violation

４．結言

本研究では，惑星間遷移軌道のロバスト軌道設計において有効な報酬シェーピングを提案した．また，その報酬を用いて推力損失下で学習を行い，推力損失が発生しない場合と比較することで，導出されたポリシーが推力損失下での制約処理において有効であることを確認した．今後は，さらなる制約処理能力の向上および，最適性の保証について検討する．

参考文献

[1] Zavoli, A., & Federici, L."Reinforcement learning for robust trajectory design of interplanetary missions", Journal of Guidance, Control, and Dynamics, 44(8), pp. 1440-1453, 2021.

[2] Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., & Madry, A. “Implementation Matters in Deep RL: A Case Study on PPO and TRPO”, International Conference on Learning Representations, 2020.

[3] Hill, A., et al.“Stable-Baselines3: Reliable Reinforcement Learning Implementations”, Journal of Machine Learning Research, 22, pp. 1-8, 2021.