

深層学習を用いた信号時相論理仕様を満たす 制御器設計に関する研究

電気電子情報通信工学専攻 電気工学コース
インテリジェントシステム領域 高井研究室 芹澤 和伸

1. はじめに

サイバーフィジカルシステム (Cyber-Physical Systems: CPS) とは、サイバー空間における情報処理とフィジカル (現実) 空間における制御対象の動作が密接に結び付いたシステムである。図1にCPSの概念図を示す。このようなシステムでは、フィジカル空間でセンシングされた情報に基づいて、サイバー空間で制御対象に与える入力を高速に計算することが求められる。近年、深層学習 (Deep Learning) を用いた学習型の制御器は、高速に制御入力を計算する有力な手法として注目されている。

他方で、従来の深層学習では教師データとの誤差に基づく損失関数が用いられることが多く、「常に障害物領域を避ける」や「指定時間以内に目標領域へ到達する」といった時間的・空間的な仕様を直接扱うことは困難であった。そこで本稿では、深層学習を用いた学習型制御器の概要と、複雑な仕様を満たすように学習させるための信号時相論理 (Signal Temporal Logic: STL) に基づく損失関数の設計について説明する。



図1: CPSの概念図

2. 深層学習を用いた学習型制御器の概要

本研究では、制御対象のダイナミクスを以下の離散時間非線形システムで定義する。

$$x_{k+1} = f(x_k, u_k).$$

ここで、 $x_k \in \mathcal{X} \subset \mathbb{R}^{n_x}$ と $u_k \in \mathcal{U} \subset \mathbb{R}^{n_u}$ はそれぞれ時間ステップ $k \in \mathbb{Z}_{\geq 0}$ における状態

と制御入力を表し、 $\mathcal{X}$ と $\mathcal{U}$ は状態空間と制御入力空間を表す。制御入力は、状態を入力として制御入力を出力する深層ニューラルネットワーク (DNN) により、 $u_k = \pi(x_k; W)$ と与えられる。ここで、 W は制御器パラメータである。初期状態 x_0 は初期状態空間 $\mathcal{X}_0 \subset \mathcal{X}$ から確率分布 $p_0(\cdot)$ に従って与えられるものとする。初期状態 x_0 および制御入力列 $u_{0:T-1} = (u_0, u_1, \dots, u_{T-1})$ が与えられると、対応する状態列 $x_{1:T} = (x_1, x_2, \dots, x_T)$ がダイナミクスに従って生成される。本研究では、状態列 $x_{0:T} = (x_0, x_1, \dots, x_T)$ を「軌道」と呼ぶ。図2は、学習型制御器による軌道生成の流れを示している。

ニューラルネットワークの学習では、「モデルの良さ」を損失関数で定式化し、その値が小さくなるようにパラメータ W を更新する。例えば、教師データとして制御入力 $u_k^{(i), \text{ref}}$ が与えられた場合には、教師データとニューラルネットワークの出力との平均二乗誤差に基づいて、以下のような損失関数を定義できる。

$$J(W) = \frac{1}{N} \sum_{i=1}^N \sum_{k=0}^{T-1} \left\| \pi(x_k^{(i)}; W) - u_k^{(i), \text{ref}} \right\|^2 \quad \text{s.t.} \quad x_{k+1}^{(i)} = f(x_k^{(i)}, u_k^{(i)}), \quad u_k^{(i)} = \pi(x_k^{(i)}; W).$$

ここで、 N は学習データ数である。このような平均二乗誤差で定義される損失関数は、値が小さいほど、教師データとニューラルネットワークの出力との一致度が高いことを表すものである。そのため、この損失関数を最小化するように学習することで、すなわち最小化するようにパラメータ W を更新することで、ニューラルネットワークは教師データに近い制御入力を生成するように学習される。

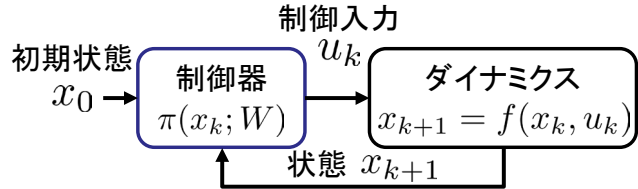


図2: 学習型制御器による軌道生成の流れ

3. 信号時相論理仕様に基づく損失関数の設計

前章で説明した平均二乗誤差に基づく損失関数は、教師データとの近さを評価する指標であるため、先述した時間的・空間的な仕様を直接扱うことは困難である。そこで、仕様そのものを損失関数に反映するために、信号時相論理 (STL) に基づいた定量的指標を導入する。時相論理とは、命題論理で

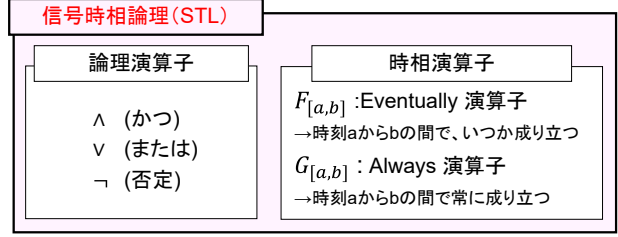


図3：信号時相論理で用いられる演算子の一例

用いられる論理演算子に時相演算子を追加した論理体系である。例えば「論理式 ϕ が真になるまで、論理式 ψ が真である」は、until 演算子 U を用いて、 $\psi U \phi$ と表される。信号時相論理 (STL) は、時相論理を実数値信号に拡張したものであり、軌道が満たすべき時間的・空間的仕様を記述できる。STL は原子命題、論理演算子、時相演算子を用いて再帰的に構成され、そのシンタックスは以下のように与えられる[1]。

$$\phi ::= \mu \mid \top \mid \neg \phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \phi_1 U_I \phi_2.$$

ここで、 $\mu : \mathbb{R}^{n_x} \rightarrow \mathbb{B}$ は述語であり、関数 $h : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ の符号に基づいて定義される。より具体的には、 $h(x) > 0$ のとき μ は真であり、その他は偽である。時相演算子 U_I は、時間区間 $I = [a, b]$ に対する until 演算子を表す。図3に示した eventually 演算子 F_I および always 演算子 G_I は、until 演算子を用いて、それぞれ $F_I \phi := \top U_I \phi$ および $G_I \phi := \neg F_I \neg \phi$ と表される。

STL の Boolean セマンティクスは、軌道 $x_{t:T} = (x_t, x_{t+1}, \dots, x_T)$ に対して以下のように定義される。

$$\begin{aligned} x_{t:T} \models \mu &\iff h(x_t) > 0, & x_{t:T} \models \neg \mu &\iff \neg(x_{t:T} \models \mu), \\ x_{t:T} \models \phi_1 \wedge \phi_2 &\iff x_{t:T} \models \phi_1 \wedge x_{t:T} \models \phi_2, & x_{t:T} \models \phi_1 \vee \phi_2 &\iff x_{t:T} \models \phi_1 \vee x_{t:T} \models \phi_2, \\ x_{t:T} \models \phi_1 U_I \phi_2 &\iff \exists t_1 \in t + I = \{t + k \in \mathbb{Z}_{>0} : k \in I\} \text{ s.t. } x_{t_1:T} \models \phi_2 \wedge \forall t_2 \in [t, t_1], x_{t_2:T} \models \phi_1. \end{aligned}$$

STL では、この Boolean セマンティクスを拡張して、定量的なセマンティクスである Robustness score が定義される[2]。Robustness score $\rho^\phi(x_{t:T})$ は、「軌道 $x_{t:T}$ が STL 仕様 ϕ をどの程度余裕を持って満たすか」を表す指標であり、以下のように定義される。

$$\begin{aligned} \rho^\mu(x_{t:T}) &= h(x_t), & \rho^{\neg \mu}(x_{t:T}) &= -h(x_t) \\ \rho^{\phi_1 \wedge \phi_2}(x_{t:T}) &= \min(\rho^{\phi_1}(x_{t:T}), \rho^{\phi_2}(x_{t:T})), & \rho^{\phi_1 \vee \phi_2}(x_{t:T}) &= \max(\rho^{\phi_1}(x_{t:T}), \rho^{\phi_2}(x_{t:T})), \\ \rho^{\phi_1 U_I \phi_2}(x_{t:T}) &= \max_{t_1 \in t+I} \left(\min(\rho^{\phi_2}(x_{t_1:T}), \min_{t_2 \in [t, t_1]} \rho^{\phi_1}(x_{t_2:T})) \right). \end{aligned}$$

ここで、 $\rho^\phi(x_{t:T}) > 0$ であれば軌道 $x_{t:T}$ は STL 仕様 ϕ を満たし、さらに大きいほど余裕を持って仕様を満たすことを意味する。本研究では、Robustness score を用いて以下のような損失関数を定義する。

$$J(W) = \frac{1}{N} \sum_{i=1}^N \left(-\rho^\phi(x_{0:T}^{(i)}) + \gamma \sum_{k=0}^{T-1} g(x_k^{(i)}, u_k^{(i)}) \right) \quad \text{s.t.} \quad x_{k+1}^{(i)} = f(x_k^{(i)}, u_k^{(i)}), \quad u_k^{(i)} = \pi(x_k^{(i)}; W).$$

ここで、 $g(\cdot)$ は制御コストを表し、過度に保守的な軌道が生成されることを防ぐ正則化項である。 γ は、Robustness score と制御コストの重みを調整する係数である。このような損失関数を最小化するようにパラメータ W を更新することで、制御器は STL 仕様を満たすような制御入力を生成するように学習される。

4. まとめ

本稿では、深層学習を用いた学習型制御器に関する概要と、信号時相論理仕様に基づく損失関数の設計について説明した。本研究に関する実験結果等については、文献[3]を参照されたい。

5. 参考文献

- [1] N. Mehdipour, et al., “Arithmetic-Geometric Mean Robustness for Control from Signal Temporal Logic Specifications,” 2019 Annual American Control Conference (ACC).
- [2] K. Leung, et al., “Semi-Supervised Trajectory-Feedback Controller Synthesis for Signal Temporal Logic Specifications,” 2022 American Control Conference (ACC).
- [3] K. Serizawa, et al., “Clustering-based Recurrent Neural Network Controller Synthesis under Signal Temporal Logic Specifications,” arXiv:2504.19846.