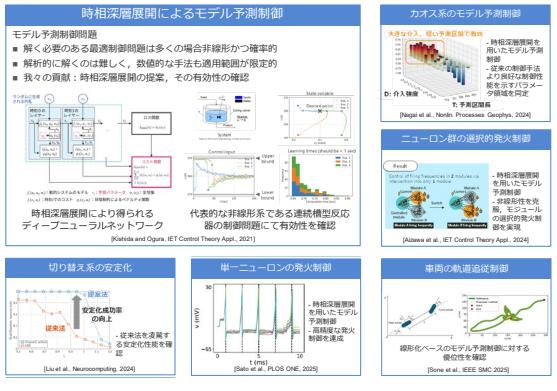


## 汎用性

一つの枠組みで多様な制御を実現

- 非線形・時変・大規模系にも適用可能
  - 制御・信号処理・機械学習を跨いだ利用が容易
  - パラメータや構造変更への柔軟な対応が可能



## 使用性

## 動くシステムを短期間で実現

- 制御対象の数理モデルがあれば数週間で完了
  - コードは極めてシンプル

```

using LinearAlgebra
using Random
using Flux
A = [0.5 0.7; 0.8 1]; #不安定システム
B = [-0.2 0.1; -0.7 -0.6];
C = [0.8 -0.7];
function loss(K, k)
    xk = (A+B*K*C)^k * rand(2, 1)
    return norm(xk)^2
end
Random.seed!(1)
opt = Flux.ADAM(1e-0)
K = zeros(2, 1);
ps = Flux.params(K)
D = 300
for i in 1:D
    gs = gradient(ps) do
        loss(K, 10)
    end
    Flux.Optimise.update!(opt, ps, g)
end

```

[小蔵, 岸田, システム/制御/情報]

## 有效性

多様なシステムで高い性能  
を実証

- AI分野のオプティマイザをそのまま利用可能
  - 多様な制御問題で有効性を確認済み
  - 従来手法に比べて高速な収束や制御成功率を達成

# 数理モデルの信頼性 とAIの柔軟性を統合 した次世代制御 フレームワーク