第2章 極配置法/線形最適制御から
オブザーバを用いた状態量の推定まで

状態フィードバック制御による制御器の設計&シミュレーション

廣川 類

リスト1　倒立振子システムのシミュレーションを行う関数を定義

```
def CartPoleSim(env, sys, ulim, F, L=None):

    nstep = 400
    t = np.arange(0, nstep+1)*env.tau

    x = np.zeros((nstep+1, 4))
    u = np.zeros((nstep+1, 1))
    xs = np.zeros((nstep+1, 4))

    xh, _ = env.reset(seed=1)
    x[0, :] = xh

    for i in range(nstep):
        # controller
        u_ = -F@xs[i, :]
        u_ = np.clip(u_, -ulim, ulim)[0]
        u[i] = u_

        if L is not None:  # observer
            ys = sys.C@xh
            dx = sys.A@xs[i, :] + sys.B@u[i] +
                                  L@(ys-sys.C@xs[i, :])
            xs[i+1, :] = xs[i, :] + dx*env.tau
        else:
            xs[i+1, :] = xh

        # get force direction (action) and force
                                magnitude (force_mag)
        action = 1 if u_ > 0 else 0
        env.force_mag = np.abs(u_)

        # apply action
        xh, reward, terminated, truncated, _ =
                                    env.step(action)

        if terminated or truncated:
            print(f'Terminated after {i+1}
                                     iterations.')
            break

        x[i+1, :] = xh

    return t[:i], x[:i, :], u[:i, :], xs[:i, :]
```

リスト2　シミュレーション結果を表示する関数を定義

```
def ShowResult(t, x, u, xs=None):

    col_t = 'bmgr'
    lbl_t = ['$x$ [m]', '$\\dot{x}$ [m/s]',
             '$\\theta$ [rad]', '$\\dot\\theta$ [
                                          rad/s]']
    plt.figure()
    for k in range(4):
        plt.plot(t, x[:, k], col_t[k]+'-',
                                    label=lbl_t[k])
        if xs is not None:
            plt.plot(t, xs[:, k], col_t[k]+'--')

    plt.grid()
    plt.legend()
    plt.xlabel('time [s]')
    plt.ylabel('states')
    plt.show()

    plt.figure()
    plt.plot(t[:], u[:, :])
    plt.grid()
    plt.xlabel('time [s]')
    plt.ylabel('control input [N]')
    plt.show()
```

本章では，倒立振子システムを対象として，状態フィードバックによりシステムを安定に制御する制御器を設計し，Pythonを用いて安定性解析や時間応答シミュレーションを行うことにより設計結果を確認します．

可制御なシステムにおいては，状態フィードバック制御により，システムの極を任意の位置に配置し，不安定なシステムを安定化することができます．また，現代制御の代表的な手法であるLQR（Linear Quadratic Regulator：線形最適制御）を用いると状態量や操作量の重みを設定するだけで最適な状態フィードバック制御ゲインを求めることができます．通常，全ての状態量をフィードバック量として用いることは難しいため，オブザーバを用いて，センサなどで検出できない状態量を推定します．

シミュレーションの準備

Gymnasiumのモデルを利用することで，倒立振子の制御の様子をアニメーション表示で確認することができます．シミュレーションを行うCartPoleSim関数を定義します（**リスト1**）．第5引数にオブザーバのゲインを指定するとオブザーバによる状態量推定が行われ，フィードバック制御に使用されます．

次に状態量ベクトル$\boldsymbol{x}$や操作量$\boldsymbol{u}$の時間応答を確認するための関数ShowResultを定義します（**リスト2**）．