Układy Sterowania Optymalnego

Politechnika Poznańska
Instytut Automatyki i Robotyki

ĆWICZENIE 10

Sterowanie SDRE dla układów nieliniowych

Celem ćwiczenia jest zapoznanie z suboptymalnym algorytmem sterowania SDRE przeznaczonym dla układów nieliniowych. W ramach wprowadzenia do ćwiczenia studenci zapoznają się z parametryzacją SDC, a następnie zaprojektują oraz zaimplementują algorytm SDRE dla przykładowego układu dynamicznego.

W ramach przygotowania do ćwiczenia należy:

→ Przypomnieć wiadomości z zakresu:

- modelowania nieliniowych układów dynamicznych,
- sterowalności i stabilności układów dynamicznych,
- projektowania sterowników optymalnych LQR.

1 Parametryzacja SDC

Rozszerzona linearyzacja, znana także jako linearyzacja pozorna lub parametryzacja SDC stanowi proces przekształcenia dynamiki układu nieliniowego do formy zbliżonej do opisu układu liniowego, którego macierze A, B, C oraz D zależne są od stanu układu[1]. Niech dany będzie układ nieliniowy w postaci

$$\begin{aligned} \dot{x} &= f(x,u), \\ y &= g(x,u), \end{aligned} \tag{1}$$

gdzie x stanowi stan układu, y jest sygnałem wyjściowym, natomiast $f(x,u), g(x,u)$ stanowią pewne nieliniowe funkcje definiujące dynamikę układu. Jeśli spełnione są następujące warunki:

- funkcja $f(x,u)$ jest przynajmniej jednokrotnie różniczkowalna,
- punkt $x_0 = 0, u_0 = 0$ jest punktem równowagi, tj. $f(0,0) = 0$,

to możliwe jest przedstawienie układu (1) w postaci SDC jako

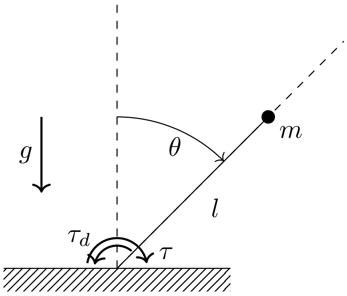
$$\begin{aligned} \dot{x} &= A(x)x + B(x)u, \\ y &= C(x)x + D(x)u, \end{aligned} \tag{2}$$

gdzie $A(x), B(x), C(x)$ oraz $D(x)$ są pewnymi macierzami zależnymi od stanu układu. Układ (2) zapisany jest w postaci zbliżonej do opisu układów liniowych. Dla tak sformułowanej dynamiki układu możliwe jest podjęcie próby zastosowania klasycznych sterowników przeznaczonych dla układów liniowych.

[1] T. Cimen, "Systematic and effective design of nonlinear feedback controllers via the state-dependent Riccati equation (SDRE) method", Annual Reviews in Control, 2010, vol. 34(1), p. 32-51

Parametryzacja SDC jest unikalna dla układów skalarnych (tj. układów o pojedynczej zmiennej stanu, $n = 1$) – dla nieliniowego skalarnego układu dynamicznego $\dot{x} = f(x)$ parametryzacja SDC przyjmuje postać $\dot{x} = \begin{bmatrix} f(x)/x \end{bmatrix} x$. W przypadku układów o większej liczbie zmiennych stanu zaproponować można nieskończoną ilość różnych parametryzacji SDC. Właściwość ta pozwala na wprowadzenie do projektu sterowania dodatkowego stopnia swobody i wpływ na odpowiedź układu zamkniętego poprzez dobór preferowanej parametryzacji dynamiki obiektu.

Niech dany będzie przykładowy układ dynamiczny w postaci wahadła pracującego w polu grawitacyjnym przedstawionego na Rys 1. Układ został przedstawiony na rysunku tak, by punkt



Rysunek 1: Układ wahadła

$x_0 = 0$ był równoznaczny z wahadłem skierowanym pionowo w górę. Przy takim założeniu, dynamika układu opisana jest równaniami

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -\frac{d}{J}x_2 + mgl\sin(x_1) + \frac{1}{J}u, \end{aligned} \tag{3}$$

gdzie $x = \begin{bmatrix} \theta & \dot{\theta} \end{bmatrix}^T$, $u = \tau$, $l = 1$m stanowi długość wahadła, $m = 9$kg określa masę kuli, $J = 1\mathrm{kg\,m}^2$ określa moment bezwładności układu napędowego, a $d = 0.5\mathrm{N\,m\,s}^2$ stanowi współczynnik tarcia. Przyspieszenie grawitacyjne dane jest jako $g = 9.81\mathrm{m/s}^2$. Dla tak zdefiniowanego układu zaproponować można parametryzację SDC w postaci

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ mgl\frac{\sin(x_1)}{x_1} & -\frac{d}{J} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix} u = A(x)x + B(x)u. \tag{4}$$

Warto zauważyć, że $\lim_{x_1 \to 0} \frac{\sin(x_1)}{x_1} = 1$ zatem poszczególne współczynniki macierzy $A(x)$ pozostają ograniczone dla dowolnych wartości zmiennych stanu x.

1.1 Przygotować funkcję `model(x,t)` implementującą dynamikę wahadła. Przeprowadzić symulację pracy wahadła przyjmując warunek początkowy $x(0) = \begin{bmatrix} \frac{\pi}{4} & 0 \end{bmatrix}$ i zerowe wymuszenie. Zwrócić uwagę na zgodność wyników symulacji z fizyczną interpretacją układu.

1.2 Zaimplementować funkcje `A(x)` oraz `B(x)` pozwalające wyznaczyć chwilowe wartości macierzy $A(x)$ oraz $B(x)$ zgodnie z parametryzacją SDC (4).

2 Sterowanie SDRE

Nieliniowy układ przedstawiony w postaci sparametryzowanej SDC może być skutecznie sterowany poprzez wyznaczenie w każdej chwili czasu sterownika LQR dla aktualnej konfiguracji układu. Uzyskany w ten sposób algorytm sterownia określa się jako sterowanie SDRE (ang. State Dependent Riccati Equation). Rozważa się układ nieliniowy z dynamiką w postaci (2). W celu stabilizacji układu w puncie $x = 0$ proponuje się sterownik ze sprzężeniem od stanu w postaci

$$u = -K(x,t)x, \tag{5}$$

gdzie $K(x,t)$ jest zmienną w czasie i zależną od stanu układu macierzą wzmocnień. Przyjmując nieskończony horyzont czasowy, kryterium jakości sterownika SDRE ma postać

$$J = \int_0^\infty \left(x^T Q(x) x + u^T R(x) u\right) dt \tag{6}$$

i prowadzi do wzmocnień sterownika (5) w postaci

$$K = R(x)^{-1} B(x)^T P(x), \tag{7}$$

gdzie parametr $P(x)$ wyznacza się na postawie algebraicznego równania Riccatiego

$$P(x)A(x) - P(x)B(x)R(x)^{-1}B(x)^T P(x) + A(x)^T P(x) + Q(x) = 0. \tag{8}$$

Macierz wzmocnień, macierz P oraz samo równanie Ricattiego zależne są od chwilowego stanu układu. W związku z tym, niezbędne jest cykliczne rozwiązywanie równań Riccatiego w kolejnych chwilach czasu, w celu wyznaczenia nowych wartości wzmocnień sterownika. Ponadto, ze względu na konieczność cyklicznego rozwiązywania równań Riccatiego, możliwe jest zdefiniowanie macierzy wzmocnień Q, R, S jako zależnych od stanu macierzy $Q(x), R(x), S(x)$. Zabieg ten umożliwia precyzyjniejsze kształtowanie odpowiedzi układu zamkniętego, np. w celu skuteczniejszej regulacji wybranych stanów układu.

2.1 Zmodyfikować funkcję `model(x,t)` definiując sygnał sterujący zgodnie z (5) oraz wyznaczając wartości wzmocnień regulatora zgodnie z algorytmem SDRE dla nieskończonego horyzontu czasowego.

2.2 Przeprowadzić symulację pracy układu. Przyjąć warunki początkowe $x(0) = \begin{bmatrix} 2\pi & 0 \end{bmatrix}^T$. oraz stałe, jednostkowe wzmocnienia Q, R, S.

- Czy zadanie stabilizacji układu realizowane jest poprawnie?

2.3 Zmodyfikować przygotowany kod, wprowadzając zależną od stanu macierz $Q(x) = \begin{bmatrix} x_1^2 & 0 \\ 0 & x_2^2 \end{bmatrix}$. Powtórzyć symulację układu zamkniętego.

- Jak zmieniły się przebiegi zmiennych stanu oraz sygnału sterującego?
- Czy zmiana przebiegów jest zgodna z interpretacją znaczenia macierzy Q?

□

Rozwiązania

```
#%% initialize constant values
l = 1
m = 9
J = 1
g = 9.81
d = 0.5

def A(x):
    return np.matrix([[0, 1], [m*g*l*np.sin(x[0])/x[0], -d/J]])

def B(x):
    return np.matrix([[0],[1/J]])

def Q(x):
    #return np.matrix([[1,0],[0,1]])
    return np. matrix([[x[0]**2,0],[0,x[1]**2]])

def R(x):
    return np.matrix([[1]])

#%% simulation time
tf = 10
t=np.arange(0,tf,0.01)

#%% LQR calculations
def control(t, x):
    P = solve_continuous_are(A(x),B(x),Q(x),R(x))
    K = R(x).I @ B(x).T @ P

    u = - K @ np.matrix([x]).T

    return u

#%% define dynamic system
def model(t, x):

    u = control(t,x)
    #u = 0

    dx = x.copy()
    dx[0] = x[1]
    dx[1] = 1/J*u - d/J*x[1] + m*g*l*np.sin(x[0])

    return dx

#%% simulate the dynamic system
res = odeint(model, [2*np.pi,0], t, tfirst=True, rtol=1e-10)

#plot results
plt.plot(t, res[:,0])
plt.plot(t, res[:,1])
plt.grid();
plt.legend([r'$\theta$',r'$\dot\theta$'])
```