Układy Sterowania Optymalnego

Politechnika Poznańska
Instytut Automatyki i Robotyki

ĆWICZENIE 6

Regulator optymalny LQR

Celem ćwiczenia jest prezentacja regulatora liniowo-kwadratowego, jego budowy oraz przykładowych zastosowań. W tym ćwiczeniu przedstawiony zostanie schemat regulacji w skończonym i nieskończonym horyzoncie czasowym. W ramach realizacji ćwiczenia studenci posłużą się językiem Python do zaimplementowania regulatora liniowo-kwadratowego, a następnie wykorzystają go w układzie sterowania.

W ramach przygotowania do ćwiczenia należy:

→ Przypomnieć wiadomości z zakresu:

- opisu liniowych układów dynamicznych w przestrzeni zmiennych stanu,
- projektowania układów regulacji (pojęcia uchybu, wartości zadanej, sprzężenia, itp.)

1 Regulator LQR z nieskończonym horyzontem czasowym

Regulator LQR (ang. Linear Quadratic Regulator) jest podstawowym algorytmem sterowania należącym do grupy algorytmów sterowania optymalnego. Algorytm wykorzystuje sprzężenie od stanu, dla którego wartości wzmocnień wyznaczone są tak, by zapewnić minimalizację kwadratowego wskaźnika jakości sterowania. Regulator LQR wykorzystuje pojęcie tzw. horyzontu czasowego – chwili czasu w której oczekuje się uzyskania minimalizacji zadanego wskaźnika regulacji. W przypadku sterownika z nieskończonym horyzontem czasowym przyjmuje się, że zadany wskaźnik osiągnie wartość minimalną dla chwili czasu $t = \infty$.

Niech dany będzie układ liniowy opisany równaniem różniczkowym w postaci

$$\dot{x} = Ax + Bu, \tag{1}$$

gdzie x jest wektorem stanu, u wektorem sygnałów sterujących, natomiast A oraz B stanowią odpowiedni macierz stanu oraz macierzy wejścia. Proponuje się sygnał sterujący w postaci

$$u = -Kx, \tag{2}$$

gdzie K jest pewnym stałym wektorem wzmocnień. Zadaniem algorytmu LQR jest odnalezienie takich wzmocnień regulatora, które gwarantują minimalizację wskaźnika jakości regulacji opisanego wzorem

$$J = \int_0^\infty \left(x^T Q x + u^T R u\right) dt, \tag{3}$$

gdzie J jest przyjętym wskaźnikiem jakości, natomiast R oraz Q stanowią stałe macierze wag których wartości własne powinny być dodatnie (tzn. macierze te powinny być dodatnio określone). Tak zdefiniowany wskaźnik regulacji ma prostą interpretację – osiąga on najmniejszą wartość wtedy, gdy dla wszystkich chwil czasu wartości zarówno wektora stanu x, jak i sygnałów wejściowych u, pozostają możliwie niewielkie. Macierze Q oraz R to macierze wag dobierane przez projektanta systemu. Ich dobór ma kluczowy wpływ na charakter odpowiedzi układu pracującego w algorytmie LQR. Zwiększanie wartości macierzy Q prowadzi do szybszej zbieżności wektora stanu x w układzie, kosztem zwiększonych wartości sygnałów sterujących u. W ogólności wybór większych wartości dla macierzy Q powoduje minimalizację wskaźnika J przy mniejszej normie stanu x w porównaniu do normy sterowania u. Wzrost macierzy R skutkuje natomiast działaniem przeciwnym – spada prędkość kompensacji zbieżności wektora x, ale sygnał sterujący u pozostaje na niższym poziomie przez cały czas pracy układu.

W celu znalezienia optymalnych wartości wzmocnień minimalizujących wskaźnik jakości stosuje się Hamiltonian. Wyczerpujący opis wyprowadzenia algorytmu LQR dostępny jest w literaturze [1]. Wykazać można, że dla układu liniowego (1) z sygnałem sterującym (2) wskaźnik jakości (3) przyjmuje wartość minimalną dla wzmocnień regulatora przyjętych zgodnie z formułą

$$K = R^{-1}B^T P, \tag{4}$$

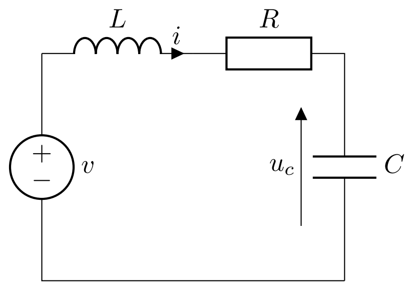
gdzie P jest pewną stałą, dodatnio określoną macierzą stanowiącą rozwiązanie tzw. algebraicznego równania Riccatiego danego wzorem

$$PA - PBR^{-1}B^T P + A^T P + Q = 0, \tag{5}$$

gdzie macierze A, B, Q, R stanowią odpowiednio macierze stanu i wejścia układu nominalnego (1) oraz macierze wag przyjętego wskaźnika jakości (3)

2 Zastosowanie

Regulator LQR może stanowić dobrą alternatywę dla regulatora PID. Niech dany będzie prosty układ elektryczny przedstawiony na Rys. 1. Przyjęto następujące wartości parametrów $R =$



Rysunek 1: Układ elektryczny RLC

0.5Ω, $C = 0.5$F, $L = 0.2$H. Przyjmując wektor zmiennych stanu w postaci $x = \begin{bmatrix} q_c & \dot{q}_c \end{bmatrix}^T = \begin{bmatrix} q_c & i \end{bmatrix}^T$ (gdzie $q_c = Cu_c$ stanowi ładunek zgromadzony na kondensatorze) oraz sygnał sterujący $u = v$ uzyskuje się reprezentację w przestrzeni zmiennych stanu

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{LC} & -\frac{R}{L} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} u = Ax + Bu. \tag{6}$$

Regulator LQR pozwala na stabilizację układu dynamicznego w punkcie równowagi, tj. $x = 0$.

2.1 Rozwiązać algebraiczne równanie Riccatiego dla dynamiki układu (6) przyjmując jednostkową macierz Q oraz $R = 1$. Do znalezienia rozwiązania równania Riccatiego wykorzystać funkcję `scipy.linalg.solve_continuous_are`. Wyznaczyć i wypisać wartości wzmocnień K zgodnie z równaniem (4)

[1] Z. Hu, L. Guo, S. Wei, and Q. Liao, "Design of lqr and pid controllers for the self balancing unicycle robot," in 2014 IEEE International Conference on Information and Automation (ICIA). IEEE, 2014, pp. 972–977

2.2 Przygotować funkcję `model(x,t)` implementującą model dynamiki układu otwartego zgodnie z równaniem (6). Funkcja powinna przyjmować na wejściu stan układu x oraz aktualną chwilę czasu t.

2.3 Przeprowadzić symulację odpowiedzi obiektu na wymuszenie skokowe w czasie $t \in (0, 5)$ s wykorzystując funkcję `odeint`.

2.4 Zmodyfikować funkcję `model(x,t)` tak, by sygnał sterujący miał postać $u = -Kx$ zgodnie z (2)

2.5 Przeprowadzić symulację układu dla niezerowych warunków początkowych. Zbadać wpływ macierzy Q oraz R na przebieg odpowiedzi układu.

- Czy macierze Q oraz R pozwalają dowolnie kształtować przebieg uchybu regulacji? Czy istnieje jakaś zależność między doborem macierzy Q oraz R?

2.6 Rozszerzyć funkcję `model(x,t)` o wyznaczanie wartości wskaźnika jakości J. Funkcja `model(x,t)` powinna wyznaczać pochodną (tj. wyrażenie podcałkowe) wskaźnika J jako dodatkową zmienną stanu – zostanie ona scałkowana przez `odeint`, a jej wartość zwrócona po zakończeniu symulacji

- Czy wyznaczona wartość rzeczywiście odpowiada minimalizowanemu wyrażeniu? W jakim horyzoncie czasu została ona wyznaczona?

3 Regulator LQR ze skończonym horyzontem czasowym

Regulator LQR ze skończonym horyzontem czasu zakłada ograniczony czas pracy układu, a poszukiwane nastawy zagwarantować mają najlepszą jakość regulacji jedynie w tym przedziale czasu. Dla układu liniowego opisanego równaniem różniczkowym (1) pobudzanego sygnałem sterującym w postaci

$$u = -K(t)x, \tag{7}$$

gdzie K jest pewnym zmiennym w czasie wektorem wzmocnień, minimalizowany wskaźnik jakości przyjmuje postać

$$J = x^T(t_1)Sx(t_1) + \int_0^{t_1} \left(x(t)^T Qx(t) + u(t)^T Ru(t)\right) dt, \tag{8}$$

gdzie J jest przyjętym wskaźnikiem jakości, S, R oraz Q stanowią stałe macierze wag których wartości własne powinny być dodatnie (tzn. macierze te powinny być dodatnio określone), natomiast $t \in [0, t_1]$ stanowi czas pracy układu. Wartość t_1 określa skończony horyzont regulacji. Interpretacja tak zdefiniowanego wskaźnika jest następująca – osiąga on najmniejszą wartość wtedy, gdy dla wszystkich chwil czasu wartości zarówno wektora stanu x, jak i sygnałów wejściowych u, pozostają możliwie niewielkie, przy czym jednocześnie wymagana jest możliwie mała wartość zmiennych stanu w końcowej chwili t_1. Macierze S, Q oraz R to macierze wag dobierane przez projektanta systemu. Ich dobór ma kluczowy wpływ na charakter odpowiedzi układu pracującego w algorytmie LQR. Zwiększanie wartości macierzy Q prowadzi do szybszej zbieżności wektora stanu x w układzie, kosztem zwiększonych wartości sygnałów sterujących u. Wzrost macierzy R skutkuje natomiast działaniem przeciwnym – spada prędkość kompensacji zbieżności wektora x, ale sygnał sterujący u pozostaje na niższym poziomie przez cały czas pracy układu. Nastawa S pozwala natomiast na zwiększanie lub zmniejszanie znaczenia jakości regulacji w chwili końcowej – jej zwiększanie prowadzi do zmniejszania wartości zmiennych stanu po zakończeniu regulacji, może jednak skutkować większymi wartościami zmiennych stanu lub sygnałów sterujących we wcześniejszych chwilach czasu. Istotne jest, iż stan uzyskany w chwili t_1 w ogólności nie jest równoznaczny stanowi ustalonemu.

Ponownie wykazać można, że dla układu liniowego (1) z sygnałem sterującym (2) wskaźnik jakości (3) przyjmuje wartość minimalną dla wzmocnień regulatora przyjętych zgodnie z formułą

$$K(t) = R^{-1}B^T P(t), \tag{9}$$

gdzie P jest pewną macierzą zmienną w czasie, dodatnio określoną macierzą stanowiącą rozwiązanie tzw. różniczkowego równania Riccatiego danego wzorem

$$P(t)A - P(t)BR^{-1}B^TP(t) + A^TP(t) + Q = -\dot{P}(t), \tag{10}$$

gdzie macierze A, B, Q, R stanowią odpowiednio macierze stanu i wejścia układu nominalnego (1) oraz macierze wag przyjętego wskaźnika jakości (3). W celu rozwiązania równania (10) konieczne jest zdefiniowanie warunku krańcowego dla macierzy $P(t)$. Na podstawie warunku transwersalności i kryterium Pontriagina wykazać można [2], że warunek końcowy

$$P(t_1) = S \tag{11}$$

zapewnia optymalizację przyjętego kryterium jakości. Należy podkreślić, że w tym przypadku znana jest jedynie wartość końcowa rozwiązania równania różniczkowego, a nie warunek początkowy.

3.1 Przygotować funkcję `riccati(p,t)` implementująca różniczkowe równanie Riccatiego (10). Zdefiniować wektor chwil czasu od t_1 do 0 przyjmując $t_1 = 5$s Wykorzystując funkcję `odeint` wyznaczyć przebieg wartości macierzy P w czasie. Zwrócić uwagę na konieczność konwersji macierzy P do postaci wektorowej dla uzyskania zgodności z funkcją `odeint`. Wykorzystać na przykład `np.reshape`, `squeeze` oraz `np.tolist`.

3.2 Wykreślić przebieg elementów macierzy $P(t)$ w czasie. Zweryfikować poprawność wyników poprzez porównanie z warunkiem krańcowym (11).

3.3 Zmodyfikować funkcję `model(x,t)` tak, by wprowadzić do niej wyznaczone wcześniej wartości macierzy $P(t)$. Wykorzystać `interpolate.interp1d` w celu określenia wartości macierzy $P(t)$ w wybranej chwili czasu (wykorzystać opcję `fill_value='extrapolate'` by zapewnić poprawną interpolację na krawędziach przedziału).

3.4 Przeprowadzić symulację odpowiedzi obiektu na wymuszenie skokowe w czasie $t \in (0, 5)$ s wykorzystując funkcję `odeint`.

3.5 Zmodyfikować funkcję `model(x,t)` tak, by sygnał sterujący miał postać zgodną z (2)

3.6 Przeprowadzić symulację układu dla niezerowych warunków początkowych. Zbadać wpływ macierzy S, Q oraz R na przebieg odpowiedzi układu.

- Czy macierze S, Q oraz R pozwalają dowolnie kształtować przebieg uchybu regulacji? Czy istnieje jakaś zależność między doborem tych macierzy?

3.7 Rozszerzyć funkcję `model(x,t)` o wyznaczanie wartości wskaźnika jakości J. Funkcja `model(x,t)` powinna wyznaczać pochodną (tj. wyrażenie podcałkowe) wskaźnika J jako dodatkową zmienną stanu – zostanie ona scałkowana przez `odeint`, a jej wartość zwrócona po zakończeniu symulacji

- Czy wyznaczona wartość rzeczywiście odpowiada minimalizowanemu wyrażeniu? W jakim horyzoncie czasu została ona wyznaczona?

3.8 Powtórzyć symulację dla $t_1 = 2$s oraz zmiennych wartości nastaw S, Q, R.

- Czy układ osiąga stan ustalony? Jaki teraz wpływ mają poszczególne nastawy?

[2] S. M. Aseev and A. V. Kryazhimskiy, "The pontryagin maximum principle and transversality conditions for a class of optimal control problems with infinite time horizons," SIAM Journal on Control and Optimization, vol. 43, no. 3, pp. 1094–1119, 2004.

4 Stabilizacja w punkcie

W zastosowaniach praktycznych stabilizacja układu w punkcie $\boldsymbol{x} = \mathbf{0}$ jest często niesatysfakcjonująca, natomiast pożądane jest przeprowadzenie układu do pewnej niezerowej konfiguracji $\boldsymbol{x}_d$, w którym powinien on zostać ustabilizowany[3]. Dla takiego przypadku rozważać można problem stabilizacji błędów regulacji w punkcie $\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{x} = 0$. Dla ogólnej postaci systemu liniowego (1) oraz zadania stabilizacji w punkcie ($\dot{\boldsymbol{x}}_d = 0$) można opisać dynamikę błędu jako

$$\dot{e} = \dot{x_d} - \dot{x} = -Ax - Bu. \tag{12}$$

Aby uzyskać poprawną postać równań dynamiki systemu, konieczne jest usunięcie zmiennej x z równań opisujących błąd. Dodając do powyższego składnik $(Ax_d - Ax_d)$ uzyskuje się poprawną dynamikę układu

$$\dot{e} = A\left(x_d - x\right) - Bu - Ax_d = Ae - Bu - Ax_d. \tag{13}$$

Postać ta różni się od postaci liniowej (1) występowaniem składnika stałego $A\boldsymbol{x}_d$. Sterowanie LQR powinno zatem kompensować ten składnik, wprowadzając odpowiednią modyfikację do sygnału sterującego w postaci $u = -u_e + u_c$ gdzie u_c dobrane jest tak, by

$$-Bu_c - Ax_d = 0, \tag{14}$$

natomiast u_e jest dobrane zgodnie z metodą LQR. Dynamika błędu przyjmuje wtedy postać

$$\dot{e} = Ae + Bu_e. \tag{15}$$

Układ ten ma postać klasycznego układu liniowego zgodną z (1). Stabilizacja układu (15) prowadzi do realizacji zdefiniowanego zadania – ponieważ $e = x_d - x$, to $e \to 0$ prowadzi do $x \to x_d$. Traktując wektor uchybów e jako wektor stanu układu, możliwe jest zastosowanie regulatora LQR do minimalizacji funkcji celu i stabilizacji układu zamkniętego poprzez sygnał sterujący zgodny z (2).

Przyjmując sygnał sterujący $u_e = -Ke$ oraz wyznaczając wartości wzmocnień K takie, by zapewniały (dla nieskończonego horyzontu czasu) minimalizację wskaźnika regulacji

$$L = \int_0^\infty \left(e^T Q e + u_e^T R u_e\right) dt \tag{16}$$

uzyskuje się sterowanie optymalne z wykorzystaniem metody LQR stabilizujące stan układu w pożądanym punkcie x_d. Zauważyć można, że dla układu zamkniętego algebraiczne równanie Riccatiego przyjmuje taką samą postać jak dla układu otwartego – macierze stanu oraz wejścia układu zamkniętego pozostają takie same jak w układzie zamkniętym. Zgodnie z zasadą superpozycji regulacja zamkniętego układu liniowego sprowadza się do stabilizacji układu otwartego w punkcie $u = 0$ oraz wprowadzeniem dodatkowego sygnału sterującego przeprowadzającego układ do pożądanego stanu zadanego.

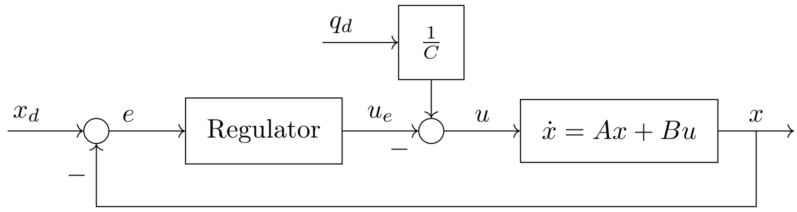
Niech rozważane będzie zadanie stabilizacji stanu układu danego rysunkiem (1) w wybranym punkcie

$$\boldsymbol{x}_d = \begin{bmatrix} x_{1d} & x_{2d} \end{bmatrix}^T = \begin{bmatrix} q_d & 0 \end{bmatrix}^T. \tag{17}$$

Należy zwrócić uwagę na fakt, że dla zmiennych stanu przyjętych zgodnie z równaniem (6) zachodzi zależność $x_2 = \dot{x}_1$ – stabilizacja układu w stałym punkcie wymaga, by $x_{2d} = 0$. Poszukuje się opisu dynamiki błędu w postaci równania stanu. Dla takiej definicji układu zaprojektowany zostanie regulator LQR gwarantujący minimalizację błędów śledzenia $\boldsymbol{e}$, a zatem zbieżność wektora stanu układu wyjściowego $\boldsymbol{x}$ do zdefiniowanego wektora wartości pożądanych $\boldsymbol{x}_d$.

Dynamika błędu ma postać (13), gdzie $Ax_d = \begin{bmatrix} 0 & -\frac{1}{LC} q_d \end{bmatrix}^T$. Ponieważ jedyny niezerowy element tego wyrażenia wpływa jedynie na zmienną sterowaną (tj. pierwszy element jest zerowy), możliwe jest jego odsprzęgnięcie poprzez zastosowanie sterowania

$$u = -u_e + \frac{1}{C} q_d, \tag{18}$$



Rysunek 2: Schemat sterowania LQR w pętli zamkniętej

gdzie u_e stanowi nowy sygnał sterujący. Implementacja układu zamkniętego przedstawiona jest na Rys. 2.

Dla rozważanego układu zaproponowane sterowanie

$$u_e = -k_1 e - k_2 \dot{e} \tag{19}$$

co jest równoznaczne z tradycyjnym sterownikiem PD. Zależność ta wynika z przyjętej reprezentacji obiektu w postaci kanonicznej sterowalnej (dla której $x_2 = \dot{x}_1$, a zatem $e_2 = \dot{e}_1$.). Zależność ta obrazuje bliski związek między różnymi technikami sterowania, w szczególności sterowaniem poprzez sprzężenie od stanu, sterowaniem LQR oraz sterowaniem PID.

4.1 Zmodyfikować funkcję `model(x,t)` tak, by sygnał sterujący wyznaczany był zgodnie ze schematem przedstawionym na Rys. 2. Wyznaczyć wzmocnienia regulatora zgodnie z algorytmem LQR, tj. (4).

4.2 Przeprowadzić symulację układu zamkniętego dla wybranej wartości zadanej q_d. Zbadać wpływ macierzy Q oraz R na przebieg odpowiedzi układu.

4.3 Analogicznie do przedstawionych wyprowadzeń wyznaczyć sterownik LQR pracujący w skończonym horyzoncie czasowym i gwarantujący zbieżność stanu do niezerowej wartości zadanej.

4.4 Zmodyfikować funkcję `model(x,t)` tak, by implementowała ona wyznaczony sterownik ze skończonym horyzontem czasu.

4.5 Przeprowadzić symulację układu zamkniętego.

□

[3]J. Hespanha, Łecture notes on LQR/LQG controller design", 2005, pp. 37

Rozwiązania

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.linalg
import scipy.signal

L = 0.2
C = 0.5
R = 0.5

A = np.matrix([[0,1],[-1/(L*C),-R/L]]);
B = np.matrix([[0],[1/L]]);

Q = np.matrix([[1,0],[0,1]]);
R = np.matrix([[1]]);

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

#%% LQR calculations
P = scipy.linalg.solve_continuous_are(A,B,Q,R)
K = R.I @ B.T @ P

#%% setpint stabilization
q_d = 2
x_d = np.matrix([[q_d],[0]])

#%% model simulation with odeint
def model(x,t):
    x = np.reshape(x,(2,1))

    #u = np.matrix([[0]]) #open loop control
    #u = - K @ x          #stabilization in the origin x=0
    u = q_d/(C) + K @ (x_d - x) #setpoint stabilization x=x_d
    dx = A@x + B@u

    return np.asarray(dx).squeeze().tolist()

ret = scipy.integrate.odeint(model, [1,0], t)
plt.figure()
plt.plot(t,ret[:,0],label='x_0')
plt.plot(t,ret[:,1],label='x_1')
plt.grid()
plt.legend()

#%% extended model calculating the cost critetion
def model_extended(z,t):
    x = np.reshape(z[0:2],(2,1))
    L = np.reshape(z[2],(1,1))

    #u = np.matrix([[0]]) #open loop control
    #u = - K @ x          #stabilization in the origin x=0
    u = q_d/(C) + K @ (x_d - x) #setpoint stabilization x=x_d
    dx = A@x + B@u

    dL = x.T @ Q @ x + u.T @ R @ u

    return np.asarray(dx).squeeze().tolist() + [np.asarray(dL).squeeze().tolist()]

ret_extended = scipy.integrate.odeint(model_extended, [1,0,0], t)
print(ret_extended[-1,2])
```